冗長型多足歩行ロボットの研究

修士学位論文
冗長型多足歩行ロボットの研究

平成１８年度
（２００７年3月）

東京海洋大学大学院
海洋科学技術研究科
海洋システム工学専攻
黒川　由崇

目次

参考文献

謝辞

付録

1．緒言

1．1　研究背景

近年，移動型ロボットは徐々に実用化され，運用場所も整地のみならず，不整地での運用を視野に入れた機体が増加してきている．ロボットを開発する目的の一つには，極限空間で人間の代わりとして危険な作業を代行させる，または人間の作業の手助けをさせるということがある．極限空間とは，原子炉の内部などの人工構造物内部のみならず，海中，宇宙空間，砂漠，雪上，などの自然界や，戦場，地雷原などの人工危険地帯，災害に見舞われた被災地なども含まれ，それら極限空間における作業は足場の悪さも相まって困難を極める．また，それら極限空間内で最も有効な移動方法は様々であり，その空間内における作業の安全性は移動方法や移動形態と密接に関係しているので，移動形態の安全性がその作業の明暗を分ける事は明白である．さらに，極限空間での立ち往生は二次災害につながる恐れがあるので極力避けたい．よって，正確かつ安定した行動を可能とする移動機構が移動型作業用ロボットには要求される．

また，近年多く開発されているのが，コミュニケーションロボットである．コミュニケーションロボットは人間と遊んだり，人間と意思の疎通を図ったり，または人間の手伝いをしたりと，人間と空間を共有することがとても多い．よってコミュニケーション型ロボットは，第一に安全性が求められる．人間と遊ぶ事を目的として製作される小型の機体の場合は，人間が機体の関節などに手を挟んだ際に，それを検知して動作を停止する接触センサーなどが装備されている．さらに，手伝いをこなす大型のコミュニケーションロボットの場合には，接触センサーに加えて，自分の位置を常に把握できる位置センサーや，障害物回避のための対物センサーが搭載されていて，活動中に人間をむやみに傷つけないようなプログラムが施されている．しかし，万一の脚部故障などによる転倒などはそれらの範疇の外である．ロボットも機械であり，モノである限り故障，破損などの事故は付き物である．そのため，それを設計，製作する人間が事故を限りなくゼロにする努力をしなければならない．よって，複数回の使用に耐えうる耐久性，誤作動を少なくするロバスト性，機体故障の際に強い冗長性，さらにはメンテナンス性，拡張性，汎用性，経済性，安全性，それらに優れたロボットの研究が今後，重要な意味を持ってくる．

１．２　研究の目的

本研究ではロボットの多脚化と，脚部破損時の歩行変化による，歩行の冗長化を考え，故障や破損に強いシステムの構築を目指す．よって，足が損傷を受けても行動を続ける生物．特に，６足の昆虫に注目し，６足歩行ロボットの作成し，そのシステムを搭載することで，歩行システムの冗長性能を検証する．

また，卒業研究では，６足歩行モデルに適用した歩行パターンを，合計で３種類（通常の歩行を１種類と，故障状態での歩行を２種類）考えた．それとともに，本研究では，これをシステムの歩行パターンとして，実機に搭載することで，多足歩行ロボットの冗長性を実証し，さらには，それを利用し，脚部の故障または破損が機体に及ぼす影響を調べ，さらなる冗長性の向上を検討する．

２．三種類の歩行パターン

本研究で作成する冗長型多足歩行ロボットは，機体の状況を自ら把握し，機体が記憶している複数の歩行方法の中から，その状況に合った歩行方法を選択して歩行するロボットである．それを可能にするために，まず複数の歩行パターンを設計する．

２．１　歩行パターン

６足歩行ロボットの歩行パターンを設計，解析するに当たって，６足で歩行する生物の歩行方法などを元に歩行方法を考えたところ，６足で歩行するロボットには主に３種類の歩行方法が適用可能である．

１つ目は，「並み足」と言う歩行方法である．この歩行方法は，主に６足の生物が低速で歩行する際の歩行方法で，全ての足の運動周期が異なっているという特徴を持っている．

２つ目の歩行方法は，「速足」注１）という歩行方法である．この歩行方法は１度に３本ずつの足を動かすことによって歩行周期が短縮され，「並み歩行」よりも歩行速度が速くなる．しかし，６足歩行生物も人間などと同じく，歩行周期を短縮させるために足を高周波で動かすほど，筋肉や関節などにかかる負担は大きくなり，歩行機構の消耗につながるという欠点がある．

３つ目の歩行は，２本の足を同時に動かしていく歩行方法である．この歩行方法は，生物の足並みには見られない歩行方法で，今回，独自に考えた歩行パターンである．この歩行パターンは，２本の足を同時に動かすため，移動中，最低でも４本の足がしっかりと接地することができ，もし脚部が１つ破損した状態でも，移動中の接地点は３点，もしくは４点となることから，安定した歩行が見込まれる．

注１）ここで注意しておきたいのは，この研究で言うところの「速足」とは，動歩行のことではなく，ただ単に「並み足」に比べて速く動く静歩行のことである．

前述のような生物の歩行方法を解析するために，本研究では各足の運動のタイミング，歩行周期，足の状態，に着目し，各歩行パターンをまとめる．そのことによって，この表の縦の系列からは各足の状態を，横の系列からは各足の歩行周期を観察することができる．

今回，設計した歩行パターンは表１に示した三通りである．

図 1：三種類の歩行パターン

今回は６足歩行の中で，比較的シンプルかつ効率の良い三種類の歩行パターンを抽出して，それらを表にプロットすることにより，各歩行パターンの特徴を調べた．この表は，縦軸（Ｌ１～Ｒ３）注2）にて脚の位置を示し，横軸は時間を示している．赤い線で区切られた範囲が，各脚の歩行サイクルを示し，表の黒い塗りの部分が脚の接地，他，白い部分が脚の上がっている状態を示している．

歩行表の一番上にある歩行パターン１は６足歩行生物が低速で歩行する際の歩行パ

ターンを模倣して設計したものである．この歩行パターンは，６本全ての足の歩行周期が全て異なっており，大変複雑な波形が生じる．しかしながら，全歩行を通して接地している足の本数が安定しているので，極端なバランスの乱れは少ないと考えられる．

歩行表にある２番目の歩行パターン２は，６足歩行生物が高速で移動する際の歩行パターンを模倣して設計したものである．この歩行パターンは，６本中３本がそれぞれ同じ周期で運動をするため，１歩の周期が他の歩行パターンの１．５倍になり，移動速度も１．５倍になる．しかし，その分だけ，バッテリーやアクチュエーター，関節への負担は大きくなることが懸念される．

歩行表で一番下にある，歩行パターン３は今回の研究のために，オリジナルで設計した歩行パターンである．一度に３本の足を上げる歩行パターン２と一度に１本の足しか上げない歩行パターン１の間をとって，一度に２本の足をあげる歩行パターンを設計した．完成した複数の歩行パターンの中で，過去の研究結果から安定性の高いモノを一つ選んで採用した．

注２）Ｌは左(Left)，Ｒは右（Right）を示し，番号の１～３は１から順番に，前脚，中脚，後ろ脚に対応している．

２．２　機体の状態

次に上記の歩行表に，機体のあらゆる状態を書き込むことによって，様々な状況下における各歩行パターンの挙動を検出し，上記で調べた三種類の歩行パターンがどのような状況で性能を発揮するかを調べた．そのため，まず，今回は各脚に起こりうる状態を「正常」「故障」「破損」の３種類に分けて，機体脚部の状態の組み合わせから機体がおかれる全状態が何通りあるのかを求めた．

その結果，機体にある６本の足がそれぞれ３通りの状態を持つので，
機体のおかれる状態は，全てで，

$3^6 = 729$[通り]

となる．

２．３　機体状態の選別

歩行表に書き込むまでも無く，歩行が完全に不可能であるとみなせる機体状態を淘汰していく．その判定基準は，本機体は静歩行ロボットであるため，正常な歩行をするた

めには最低でも未故障の足が４本必要であり，機体が正常に立脚するためには最低でも３本の脚が必要となる．よって４本以上の脚部が異常をきたしている状態のパターンは全て淘汰する．という点と，１方向に機体が大きく傾き，そのために正常な足が持ち上がり，接地している正常な脚が４本に満たなくなると歩行は不能になるので，停止状態において，既に機体が大きく傾いているものをさらに淘汰する．という２点とする．

上記の条件１，２に従い，機体状態を選別していった結果，

７２９－不必要なパターン（５９０）＝１３９　［通り］

というように，調査対象に当てはまる機体の状態は１３９通りにまで減少した．その後，各機体状態に照らし合わせながら，歩行表の各足の系列に「故障」，「破損」などの機体状態を書き込んでいく．

また，図１の各状態（縦の系列）に合わせて，６脚歩行機の模型（後述）を動かしながら各状態のバランスや傾く方向などを判定し，さらに機体の歩行バランス，傾き判定の総計から，各歩行パターンの歩行状態を判定した．

前項の歩行解析表の詳細は図２に示す通りである．

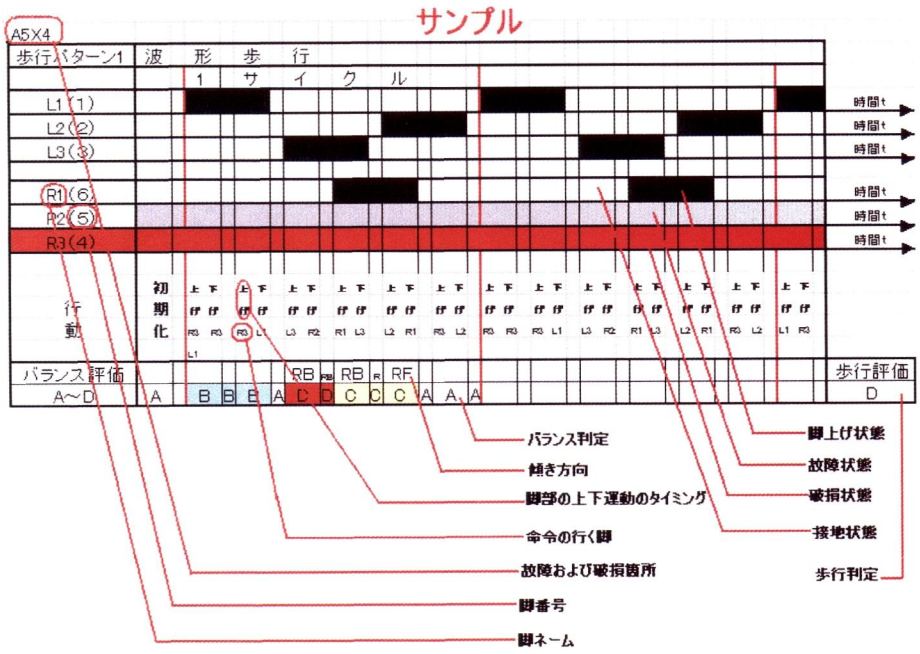


図 **2**：歩行表の詳細

図２では，さらに機体の置かれる状態を，故障状態は灰色，破損状態は赤色で塗りつぶすことによって，機体が故障や破損の際の歩行状況を見る事が可能である．上記で示している歩行解析表サンプルは，右中足（R2）が故障，右後ろ足（R3）が破損している

状態である．

次に縦の系列を見る．上記の歩行解析サンプル表の初期状態における縦の系列を見ると，初期状態において，本機体は故障中の右中足（R2）と破損している右後ろ足（R3）を除いて，全てが正常に立脚していることが解る．また，一つ時間を進めて，次の状態を見る．２つめの状態においては，故障中の右中足（R2）と破損している右後ろ足（R3）に加えて，左前足（L1）が歩行している．そして，このような歩行表の縦系列の各状態に合わせ，前述の６足歩行モデルの状態を変えて，縦系列の一番下にある欄のバランス判定を評価する．

歩行解析表は６脚歩行機の各脚の状態を行動に照らし合わせて整理し，表にしたものである．この表を利用する事によって，一見複雑そうな６脚歩行機の脚の動きをより簡単に解析することができる．

バランスの判定は，前項で述べた歩行解析表のグラフに合わせて，図３の６足歩行モデルの姿勢を，歩行解析表に描かれている波形に基づいて変えていき，その時々のモデルの状態から，機体のバランスを評価する．

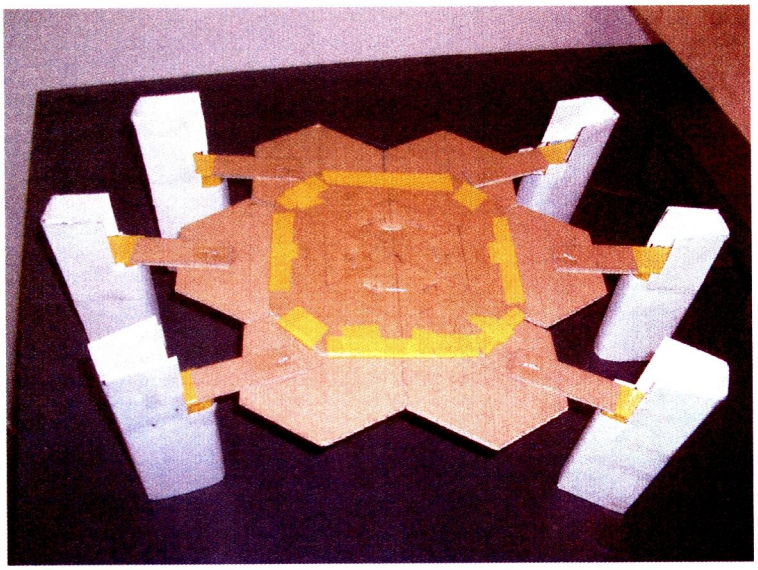


図 3：６脚歩行モデル

２．４　歩行解析シミュレーション

これまでに述べてきた歩行解析表，６足歩行モデルなどを用いて歩行の解析シミュレーションを行う．その概要を以下に示す．

２．４．１ 解析目的

今回の実験の目的は複数の歩行パターンを比較することによって，安定で故障に強い歩行方法を抽出し，より安定した歩行の実現を模索する．また，今後の実機製作のために，様々な機体状態における各歩行パターンの歩行データを収集する．

２．４．２　解析結果

３種類の歩行パターンに１３９通りの機体状態を照らし合わせ，機体のバランスと傾きを評価し，それらの総計より，機体のふらつき，歩行状態を評価した．その結果は，表１に示す．

表 1：評価結果

ふらつき判定	判　定			合計
	無い	少ない	多い	
歩行パターン1	3.6%	4.3%	92.1%	100%
波歩行	5	6	128	139(通り)
歩行パターン2	13.6%	25.2%	61.2%	100%
3脚歩行	19	35	85	139(通り)
歩行パターン3	5%	18.7%	76.3%	100%
2脚歩行	7	26	106	139(通り)

歩行判定	判　定				合計
	A	B	C	D	
歩行パターン1	2.9%	12.2%	38.8%	46.1%	100%
波歩行	4	17	54	64	139(通り)
歩行パターン2	2.2%	37.3%	32.4%	28.1%	100%
3脚歩行	3	52	45	39	139(通り)
歩行パターン3	5.8%	14.4%	56.1%	23.7%	100%
2脚歩行	8	20	78	33	139(通り)

ふらつきの判定は「無い」，「少ない」，「多い」の３種類に分類した．ふらつきが「無い」とは，歩行１サイクル中に１度も機体が傾かなかったものである．「少ない」とは，歩行周期１サイクル中に１度だけ傾く状態がある，もしくは１度傾いてから，しばらく同じ方向に傾き続けたものである．その他の複数回の傾きや転倒状態を持つものは全て「多い」と判定した．また，歩行判定は，Ａ～Ｄの４段階で評価したバランス評価をポイ

ント化し，歩行の１サイクル中のバランス評価を平均化して，歩行評価とした．しかし，例外的に１サイクル中のバランス評価において，評価Ｄをもつ歩行に関しては，転倒したという扱いにし，歩行評価もＤとした．

表１より，ふらつきが最も少ない歩行パターンは群を抜いて歩行パターン２の３脚歩行であると判定できた．しかし，歩行判定の表を見ると，歩行パターン２は様々な状況下において安定した歩行に最も適していない事がさかる．しかし，ふらつき判定で２番目に優秀な結果を残した歩行パターン３の２脚歩行は，歩行判定においても優秀な結果を残しており，さらに，前述の歩行表を見ると，歩行周波数（脚の速さ）の点で，歩行パターン２は歩行パターン１や歩行パターン３と比べて歩行周波数が１．５倍多くなり，理論的に歩行速度もそれらの１．５倍になることがわかる．しかし，部品の磨耗，電源の消耗，アクチュエーターの発熱量の点から考えると二番目に安定している歩行パターン３の歩行方法は対故障性や状況適応性の点で最も優れた歩行パターンではないかと考えられる．

本研究では，この結果を実機に搭載し，ロボットの冗長化を検討する．

３．実機制作

３．１　機体概要

本研究が，機体を故障させて始めて結果を得る研究なため，機体本体は，分解，修理が容易でありながらも，頑丈である事が望まれる．そのため，素材は主にアルミとし，図４に示すように脚部や胴体の各所に市販のブロックなどを採用し，分解や接続，修理などがより簡単に行えるように設計する．

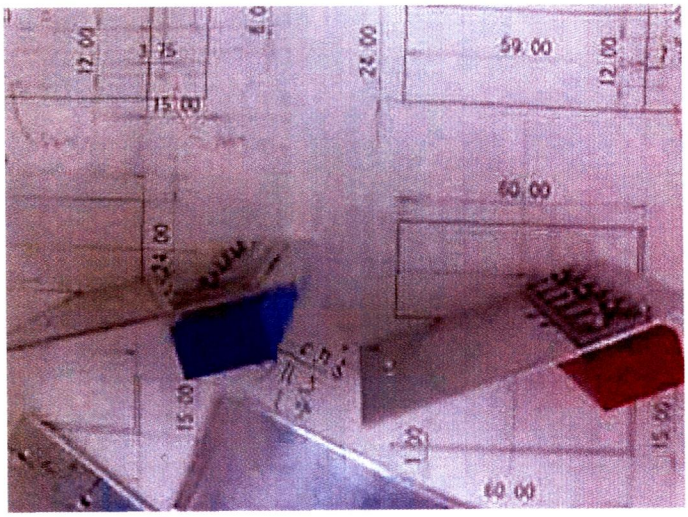


図 4：ブロックの使用例

３．１．１　脚部の設計

脚部は，機体のメンテナンス性と生産性を考え，まったく同一の脚を６つ製作する．そして，その脚を左右に配置するため，脚はできる限り左右対称になるように設計する．構成は図５に示す．脚先の関節が受動関節のため，脚部の踏ん張りは，足の設計に依存する．本機は，脚部を６本有するため，隣接する脚と脚が絡まないように設計する．

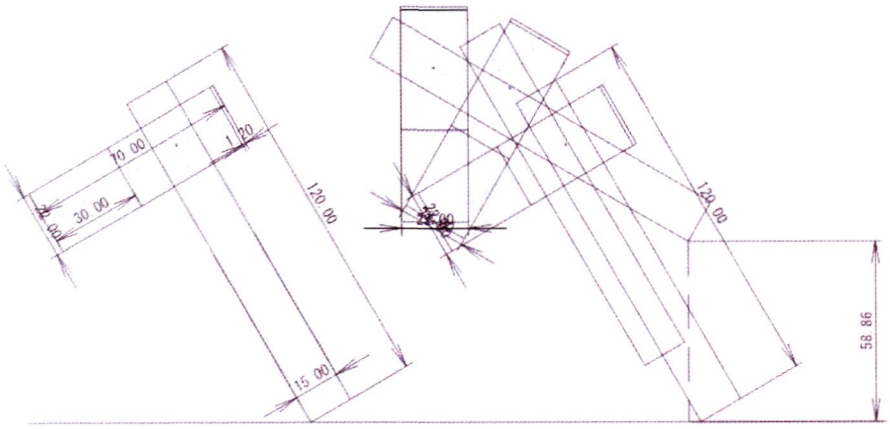


図 5：脚部の設計図

そして，完成図は図 6 となる．この脚はブロックと大型のネジによって，胴体と接続される．

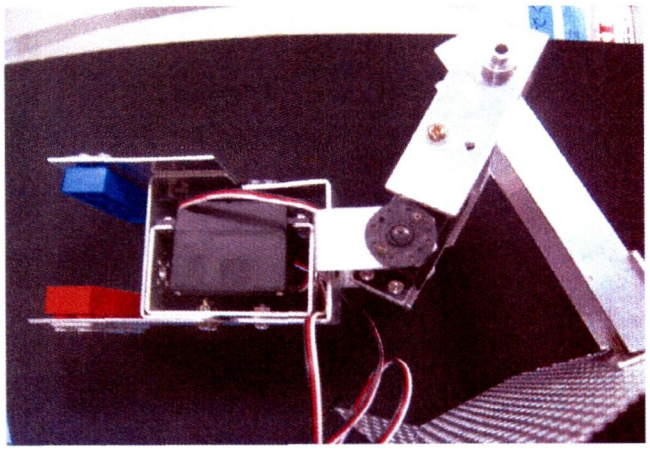


図 6：脚の完成図

３．１．２　アクチュエーター

図 7 に示すように，アクチュエーターは，ラジコンサーボモーターである．ラジコンサーボモーターは，モータードライバーとなるＰＩＣマイコンから出力するパルス信号を，変調させることによって動作する．このパルス変調制御の詳細は，制御アルゴリズムの項目で後述する．

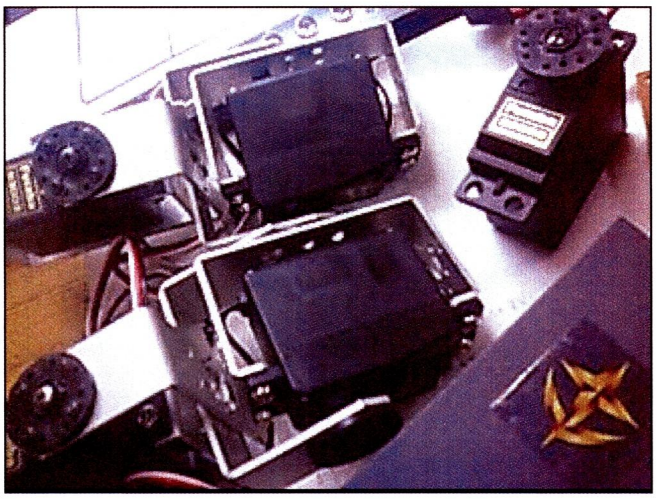


図 7：ＲＣサーボモーター

３．１．３　胴体の設計

胴体の素材はほとんどがアルミニウムである．このため，重量が 1.9kg となる．歩行の際には，この上に，さらに，基盤を装備し，重量は 3.77kg となる．そのため歩行に必要なモータートルクが不十分に出ず，歩行の際には，スタビライザー（補助器）が必要となる．しかし，図 8 に示すように静止状態では，安定した立脚を保つことができ，スタビライザーは，あくまでモーターの負担軽減が目的である．そのため，動力となる機器は何も装備しない．

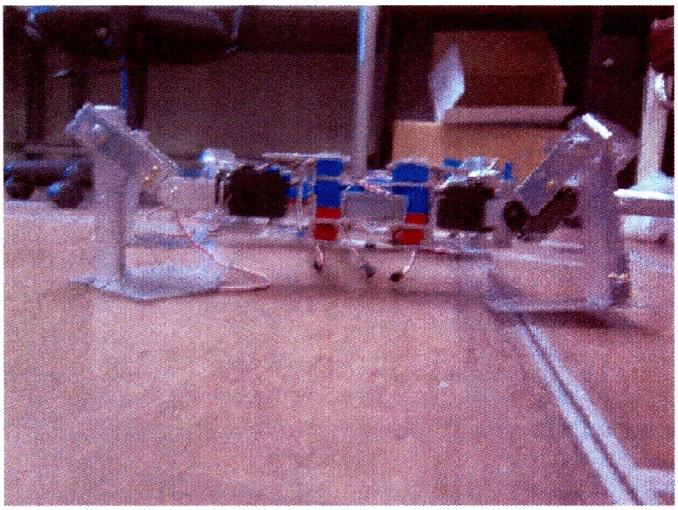


図 8：体，正面から

胴体のデータは以下のようになった．隣接する脚どうしが，歩行時に重ならない距離

をとるため，全長が 560mm となる． 機体を上から見ると，隣接する脚部が十分に離れている事が確認できる．

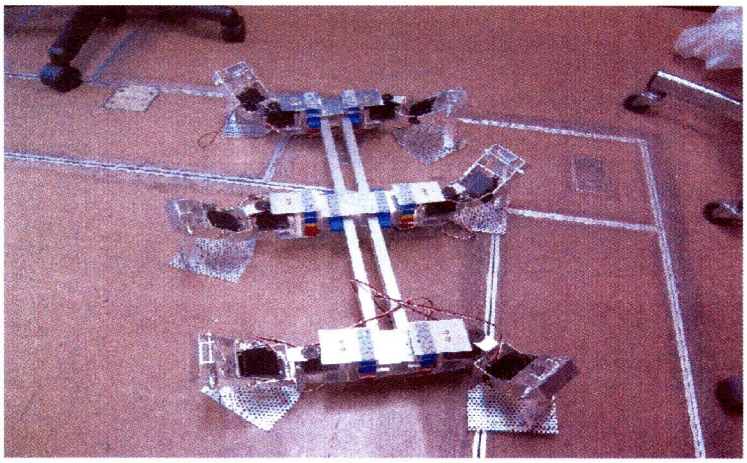


図 9：体，上から

3．1．4　足の設計

足は図 10 に示すように，接地圧と立脚のバランス，重さ，さらに，歩行中の足同士の接触による破損などを考慮し，足にはプラスチック製の板を使用し，脚の裏には，ウレタン素材のスポンジを用いている．

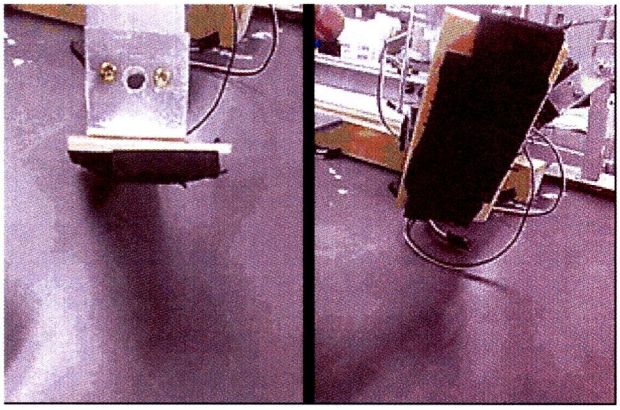


図 10：足

3．1．5　スタビライザー

図 11 の左がスタビライザーである．右が全ての基盤とバッテリーを装備した機体で

ある．

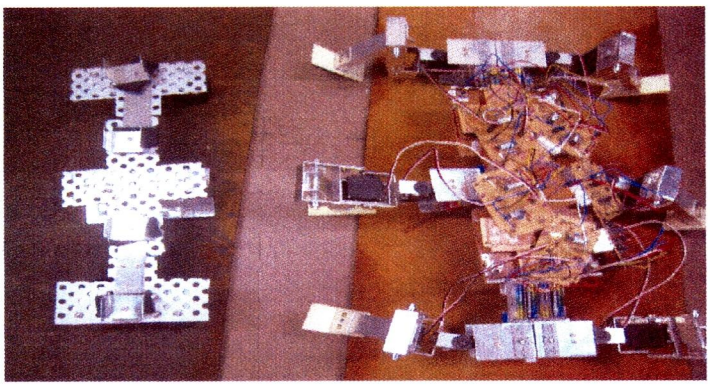


図 11：スタビライザーと本体の対比

このスタビライザーは，重くなりすぎた機体を支え，歩行をスムーズに行わせるための部分であり，アクチュエーターや動力源などは，一切装備されていない．虫の腹底部のような役割をする．スタビライザー底面には，図 12 に示すボール型のキャスターが装備されている．

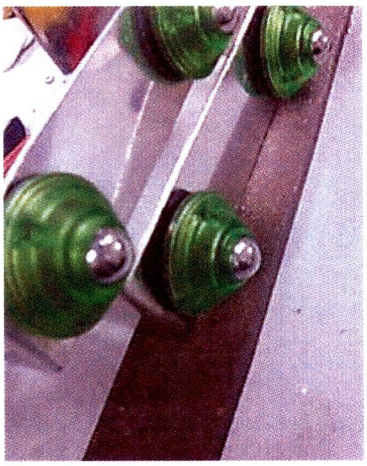


図 12：ボール型キャスター

ボール型のキャスターは，機体の運動方向に対して，タイヤ型のキャスターに比べて拘束力が小さいため，機体の運動に対する影響は少ないと考えられる．

３．１．６　電源

機体の電源は，ＰＩＣ系統と，モーター系統の２系統を用意する．これは，ＰＩＣとモーターにかかる電力がまったく異なるためである．また，モーターの動作によるノイズがＰＩＣの動作に影響を及す可能性があるためでもある．

モーター用電源には，図 13 に示す８．４Ｖ（500mA/h）のミニバッテリーを採用す

る．また，ＰＩＣ用の電源には，図 14 に示す単三電池４本（1．5Ｖ×４）を採用する．

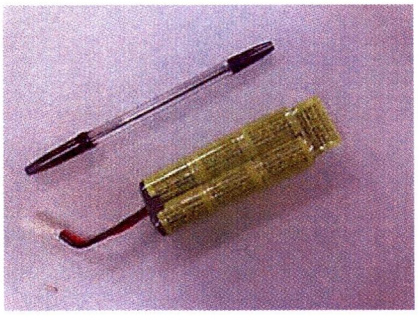


図 13：バッテリー

図 14：電池ボックス

３．１．７　機体全景

これらのスタビライザーや電源，機体の制御基盤などを装備した機体は以下のようになる．

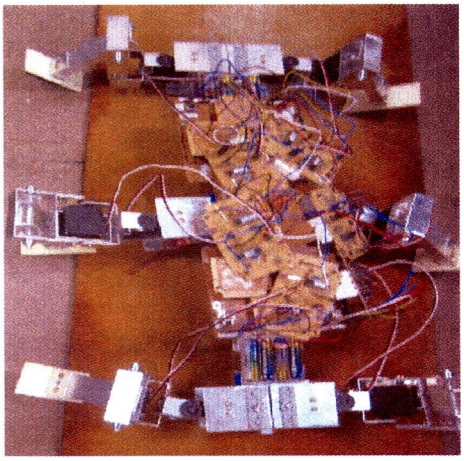


図 15：フル装備状態

機体の仕様は以下の通りとなる．

重量：　3.77kg
全長：　560mm
全高：　120mm
制御器：　ＰＩＣ(16F819, 16F873)
PIC 用電源：　単三電池（6V）
モーター用電源：　バッテリー（8.4V）
アクチュエーター：ＲＣサーボモーター(Futaba 製)

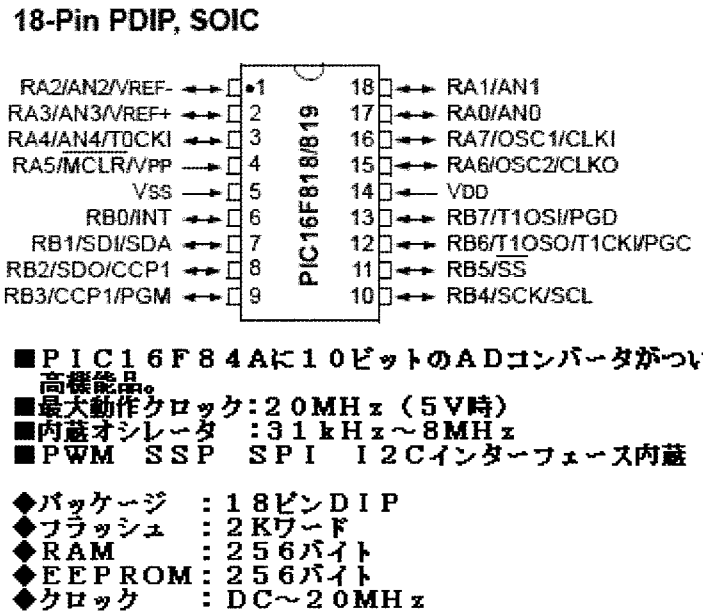
３．２　制御系統

本研究において，ロボットの制御には，ＰＩＣマイコンを使用することから，制御系統は，ＰＩＣとこれを運用する電子回路で構成されている．

３．２．１　PIC

このＰＩＣマイコンは，主に，端子のＯＮ，ＯＦＦの間隔を制御して，各機器と信号をやりとりするもので，現在では，自動車の内部部品などの製品にも使用されているマイコンである．

今回は，PIC16F819 と PIC16F873 という２種類のＰＩＣを使用する．PIC16F819 は，18 ピンのマイコンで，内部にタイマーを装備している事から，本研究ではモータードライバーなどに使用している．仕様およびピンの配置は図 16 に示す．



■ＰＩＣ１６Ｆ８４Ａに１０ビットのＡＤコンバータがついた
　高機能品。
■最大動作クロック：２０ＭＨｚ（５Ｖ時）
■内蔵オシレータ　：３１ｋＨｚ～８ＭＨｚ
■ＰＷＭ　ＳＳＰ　ＳＰＩ　Ｉ２Ｃインターフェース内蔵

◆パッケージ　：１８ピンＤＩＰ
◆フラッシュ　：２Ｋワード
◆ＲＡＭ　　：２５６バイト
◆ＥＥＰＲＯＭ：２５６バイト
◆クロック　　：ＤＣ～２０ＭＨｚ

図 16：PIC16F819 概要

PIC16F873 は，28 ピンのマイコンで，ピンの数が多い事から，多くのマイコンと信号を送受信する，Master や Grand Master の部分として使用している．PIC16F873 のピンの配置および，仕様は，図 17 に示す．

◆28P-DIP(スリムタイプ)
◆FLASH :4K14ビットワード、プログラムメモリー
◆RAM :192バイト、データメモリー
◆EEPROM:128バイト、データメモリー
◆CLOCK :DC～20MHz
◆最新のAKI-PICプログラマー(付属CDR)は対応済み

図 17：PIC16F873 概要

PIC へのプログラミングは，Micro chip 社のフリーソフトＭＰＬＡＢを使用する．プログラム開発画面を図 18 に示す．

書き込みは，秋月電商のライターソフト AKI ライターソフトと，それに対応した PIC ライターキット図 19 を使用する．用法は，ＭＰＬＡＢにより製作したプログラムをライターソフト，およびライターキットによって，ＰＩＣにロードする．

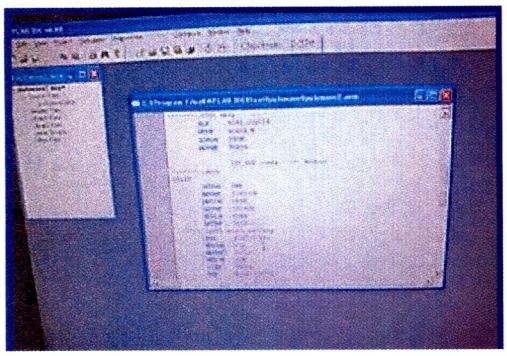


図 18：ＭＰＬＡＢ画面

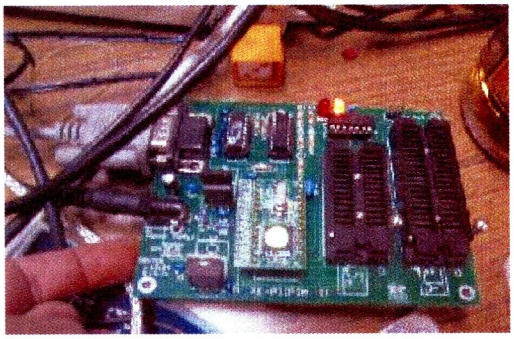


図 19：AKI　PIC ライターキット

３．２．２　電子回路設計

今回は，メンテナンス性にも配慮する観点から，電子回路を分解可能に設計する．そのため，システムの階層ごとに Slave，Master，Grand Master，Sub Master の分類で，それぞれ，異なった基盤を制作し，それを連結する基盤を介し，それら全てが繋が

るようにする．また，複数の基盤を実機へ搭載するという事も念頭に置き，基盤の１つ１つは，図 20 のように，なるべくコンパクトになるように設計する．なお，本項に出てくる Slave ， Master，Grand Master，Sub Master の詳細は，後述とする．

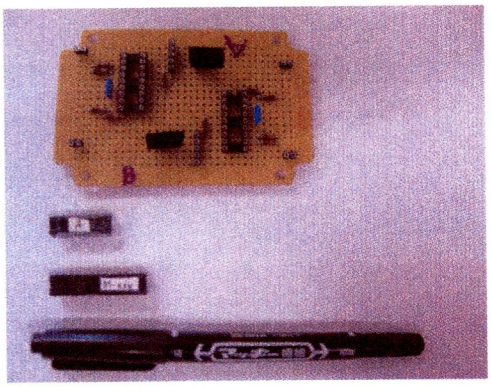


図 20：PIC と基盤の対比図

３．２．３　基盤設計

本研究で制作しているロボットの制御手法は，階層構造を形成しており，制作する基盤も，そのシステムの階層の役割ごとに異なっている．システムの構成は図 21 に示す．この機体は 12 個のサーボモーターによって，稼動している．

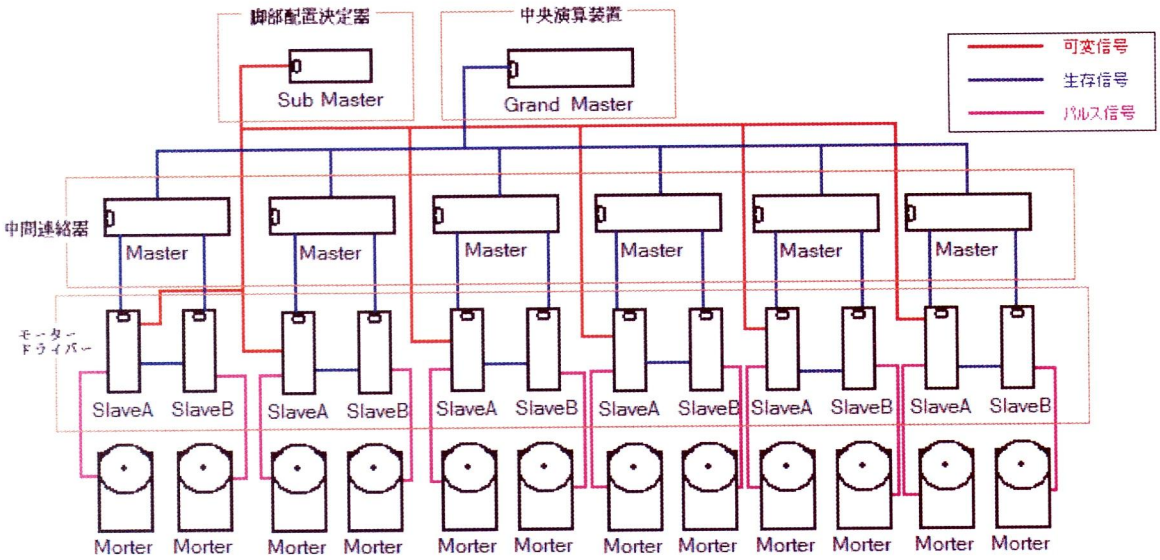


図 21：システムの構成図

このシステムの特徴は，複数のマイコンの同期をとり，機体の故障を検知することで，歩行パターンを制御する手法において，通信手法として有名なＩ２Ｃ通信を用いないこ

とである．

Slave の A，B は，同じ基盤設計とした．電源供給部分や，信号送受信部をピン端子とすることで，外部との接続を容易とした．本体とは，ピン端子付のジャンプワイヤーと言う，簡易に取り外し可能なワイヤーで配線する．

Master は，外界センサーを接続可能にするためのセンサー入力端子部分が付いているが，今回は，センサーは搭載されていない．機体本体に搭載するときには，この基盤を小型カートリッジとして機体に接続することができる．Master 基盤は図 22 に示す．

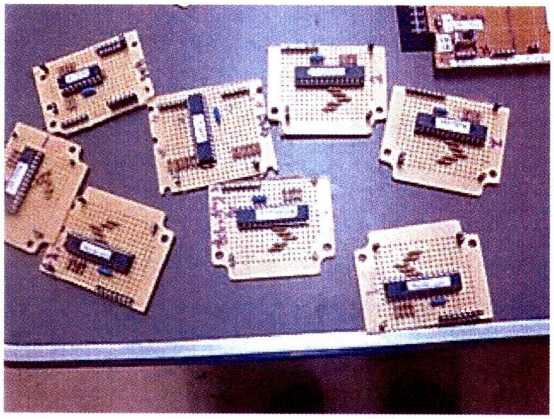


図 22：実際の Master 基盤

Grand Master は Master 同様，機体本体に搭載するときには，この基盤を小型カートリッジとして機体に接続することができる．６つの Master と，信号をやりとりするため，28 ある入出力端子をほとんど使用している．

本体基盤は Master を６枚と，Grand Master，sub Master をそれぞれ１枚にまとめて接続するための基盤．モーター用電源や，各基盤へ送る電源もこの基盤に装備されている．回路図は，以下のように，複数の基板をつなぐようになっている．

Master，Sub Master，Grand Master のコネクタと対応したピン配置を施し，
それぞれの基盤とのコネクティングを簡易に行う事ができるように設計する．

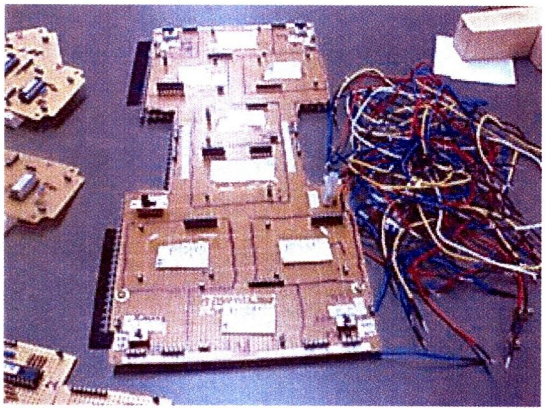


図 23：実際の本体基盤

接続は図 24，図 25 のように，基板上に設置されている凹ピンに，それぞれ対応する凸ピンを差し込むだけである．

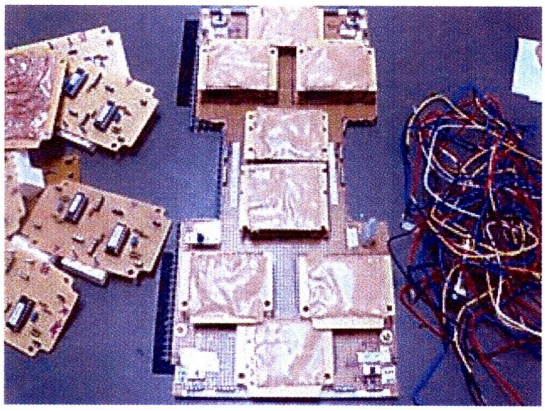


図 24：小型基盤を装着時の本体基盤

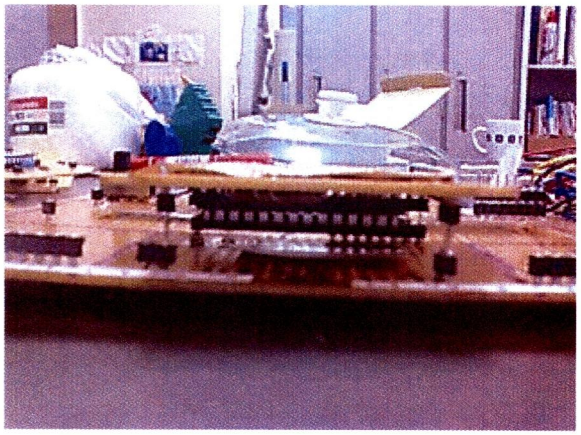


図 25：基盤，装着図

３．２．４　PIC の同期化

このロボットは，Grand Master 部分からの信号によって，足どうしの同期をとるため，足同士を統括する Master 部分の PIC は正確な同期を必要としない．しかし，２つの PIC によって，２つのサーボモーターを制御する，Slave 層の SlaveA と SlaveB は，相互に協調しながら動作することによって，足の歩行動作を形成するので，この２つの同期がずれると，足並みが乱れ，安定した歩行を行うことができなくなる．そこで，モータードライバーとなる２つの PIC の同期化が，不可欠と言える．しかし，それぞれの PIC に接続する振動発信子は，セラミック振動子（以下，セラロック）で約±0．5%，水晶振動子で約 0．000001%の誤差を持つ[9]．このようなクロック周波数の誤差は累積され，SlaveA と SlaveB の同期を狂わせ，機体の足並みを乱す原因となる．

今回，ロボット制御用の PIC には 10MHz のセラロックを用いた．このセラロックは，PIC の発信端子(OSC1)から信号を受け，クロック信号を作り出し，PIC の受信端子（OSC2）へと，信号を返すと言う原理に基づき，１秒間に１００万回の動作周期を作り出す．

図 **26**：セラロック

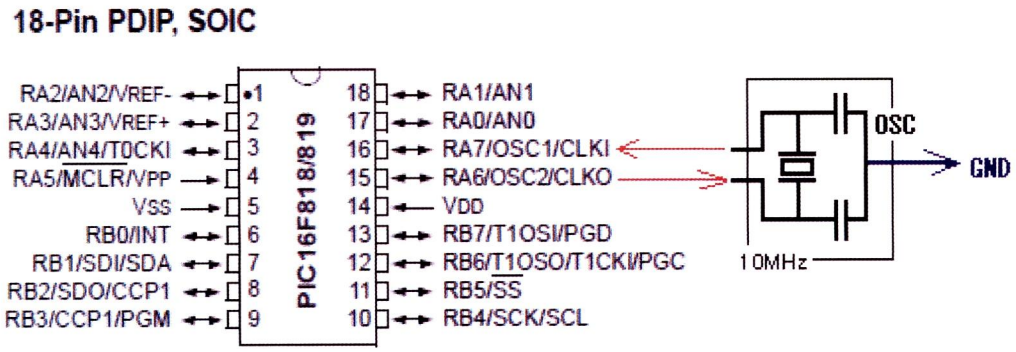


図 **27**：振動子の接続図

よって，クロックの精度が±0．5%の誤差を持つとすると，１秒間に 5000 カウント分の誤差が生じてしまう．これにより，２つの PIC をプログラムのカウント数のみで，連動させてモーターを制御しようとした場合，図 28 に示すように，それらから生成されるパルス波にずれが生じ，時間を置くと，足の動作がバラバラになってしまう．よって，複数の PIC を連動させてモーターの同期を取る目的で使用する際には，このような精度差が大きな問題となる．

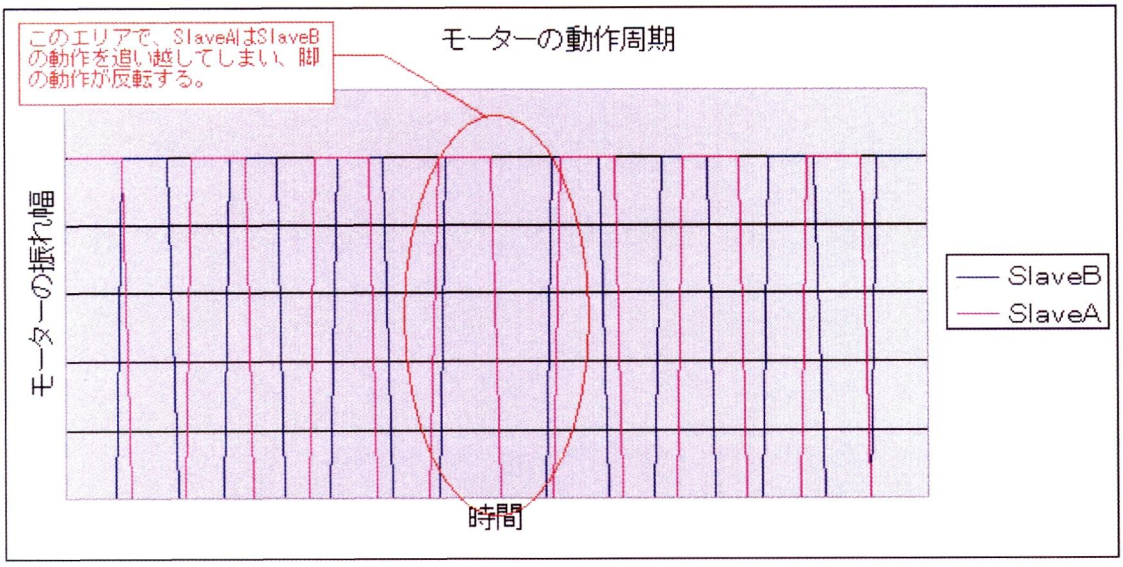


図 28：モーターへのパルス信号のずれ

そこで，今回は，２つの PIC をより正確に連動させるために，図 29 に示すように，１つのクロックから２つのクロック出力を取り出して使用するという手法によって，SlaveA と SlaveB のクロック信号を同期させるという手法をとった．この手法は，２つの PIC に１つのセラロックを共有させるので，１つのセラロックが破損すると２つの PIC が動作不能となってしまうという点では冗長性を欠いてしまうが，１つにする事でセラロックでも，２つの PIC の動作を確実に同期させる事が可能となり，また，セラロックの故障の際には，SlaveA，SlaveB の動作が同時に止まる事で，脚部２つのモーターも同時に停止し，暴走抑止にも繋がることが考えられる．

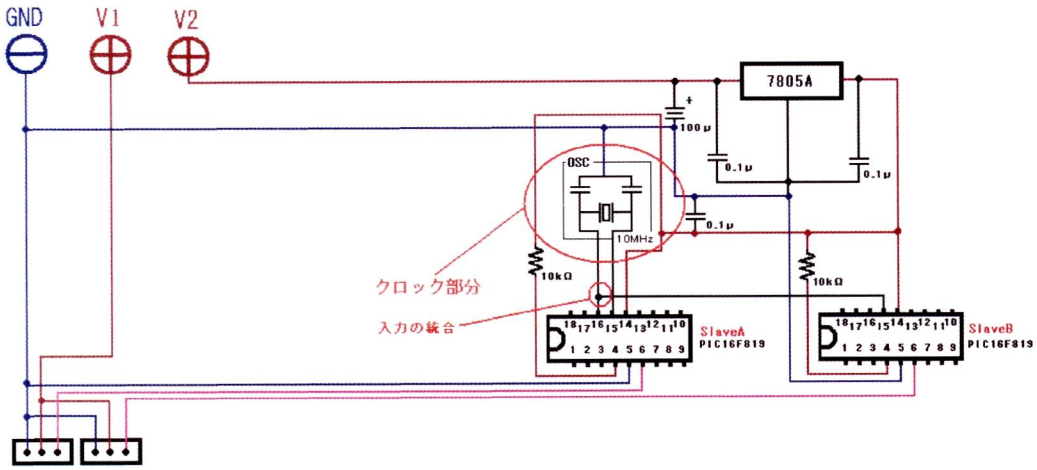


図 29：クロック同期テスト回路

本研究におけるセラロックの接続は図 29 に示すように，SlaveA が入出力するクロッ

クをマスターとなりセラロックにクロック信号を入出力し，そこから，入力信号だけを持ってくる形で，SlaveB が動作している．このことにより，SlaveA と SlaveB は，図 30 に示すように，同期を取ることが可能となった．

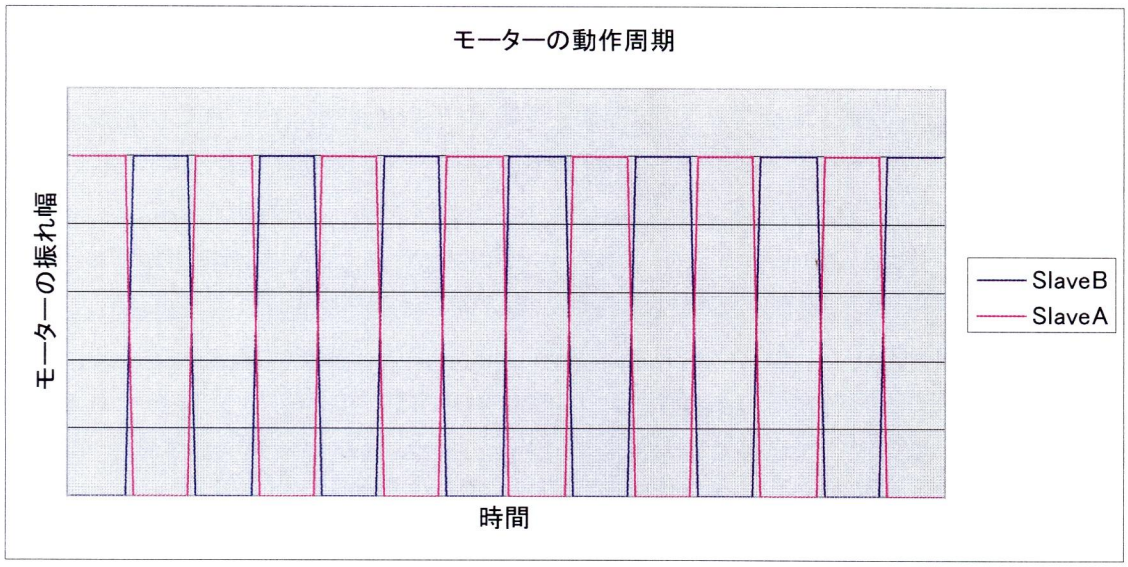


図 30：パルス波の位相差グラフ

３．３　システムの詳細

３．３．１　システムの構成

本システムは図 21 に示すように，複数のマイコンを役割ごとに階層化して，脚部の異常を検知することにより，ロボットの可変歩行を行わせるロボットシステムを開発する．通信には，通信手法として有名な I2C 通信を用いず，各 PIC から発信する ON，OFF の信号（生存信号）を相互に送信しあい，その信号が途絶える事により，PIC は周辺の異常を検知し，動作を変えるというシステムである．このことにより，I2C 通信では，マスターとスレーブという２つの階層しか作り出せないが，本システムでは Grand Master および，Sub Master，の最上位部と，Master の中間層，SlaveA，SlaveB の最下層という３つ階層を作り出すことができ，中間層が 12 個あるモータードライバー部の連絡を半分に取りまとめる事により，最上位の計算負担を軽減することができる．

３．３．２　中央電算装置（Grand Master）

中央電算装置（Grand Master）は，初期状態で中間連絡器（以下，Master）に停止信号を発信している．そして，この中央電算装置（以下，Grand Master）は Master からの生存信号を検知し，各脚の生存状況に合わせて，歩行パターンを変化させ，各脚の Master に送信している停止信号のタイミングをずらして解除していくことによって，歩行を制御する．そして，歩行中も，下部からの生存信号よって，脚の生存状況を検知し，歩行パターンの選択をおこなう．

Grand Master の動作は，図 31 に示したようになる．動作１で，下位に停止信号を送信，モーターの動きを止め，動作２で下位が完全に立ち上がるのに十分であろう時間，約２[msec]待機する．この時間は，最下層のモーターの動作周期に合わせた値である．

動作３は，初期状態で下位の生存信号の本数をカウントする．そして，動作４においてカウントした値に１を加え，その値を減算し，演算結果がゼロになった所で，歩行を変化させるサブルーチンにジャンプし，歩行が決定される．また，カウント値は生存信号の本数に依存するので，最小で０，最大で６となり，万が一，それ以上の信号が検知された際には，緊急停止信号が全脚に送信される．

この４種類の歩行のうち３種類は前述の，３種類の歩行パターンである．残る１つは，脚部が３本以下の本数になり，歩行不能となった際に，暴走を防ぎ，その場で停止するための停止信号を送信するループである．動作５は歩行パターンに伴い，各脚への停止信号をずらして解除するというループであり，walk0，walk4，walk5，walk6 という４種類のループ名が付けられている．walk のあとに付いている数字は，それぞれの歩行脚の本数に因んでいる．walk0 は，脚が３本以下の時の停止信号送信ループ．walk4 は，脚が４本になった際の波歩行を作り出すループ．walk5 は脚が５本になった際の２脚歩行を作り出すループ．walk6 は，全脚部が正常な際の３脚歩行を作り出すループである．各 walk ループについての詳細は後述する．

動作６は再び下位の生存信号をカウントするループである．このように，カウントを続け，動作７では，カウントが変化した際に再び，４種類の歩行パターンの中から歩行方法を選択する．そして，歩行の変化の際には，動作８で変化を示す Flag を立てて，動作の変化を確認する．フラグが変更されていたら，動作５に移動し，同時に変化しているカウント値に従って，歩行を変化させる．フラグが変更されてなければ動作７に行き，脚部の異常をカウントし続けながら，これまでの歩行を継続させる．

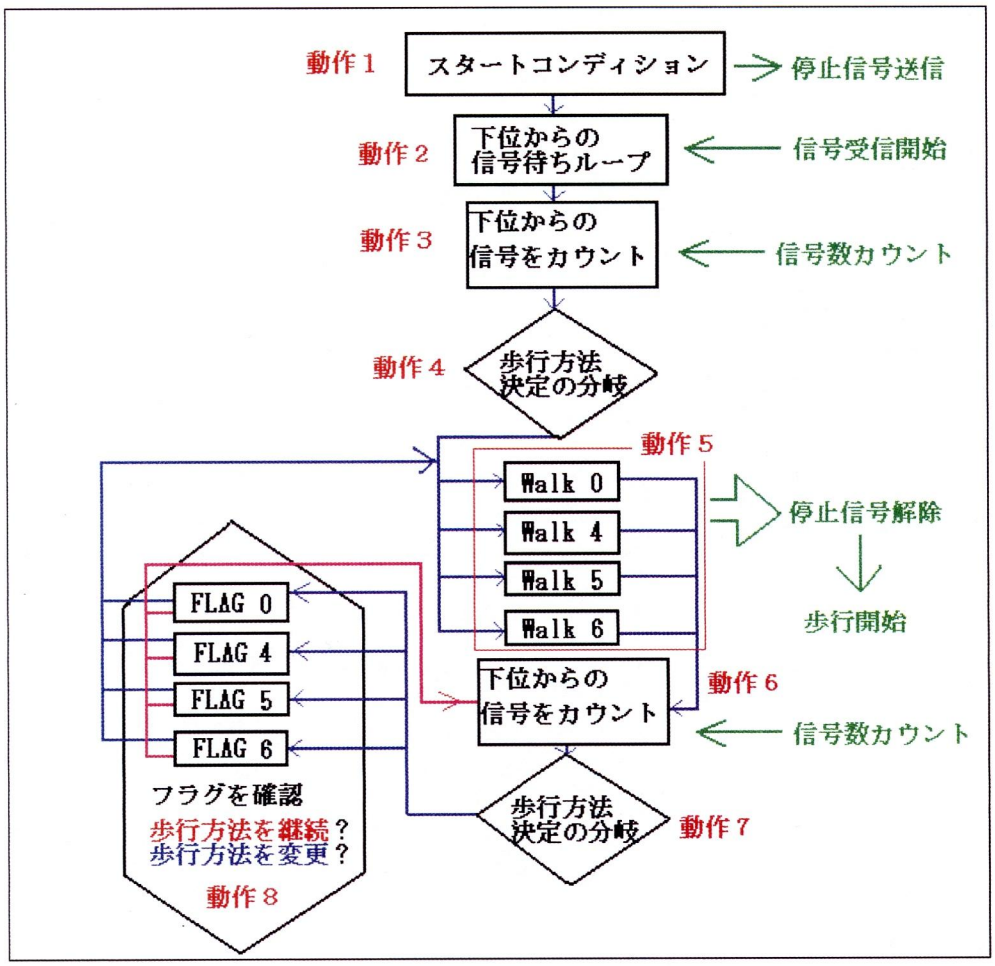


図 31：Grand Master のフローチャート

３．３．３　中間連絡器（Master）

中間連絡器（Master）は，各モータードライバー（SlaveA，B）からの生存信号を受け，上位である Grand Master に，生存信号を送信し，Grand Master から停止信号が来た場合は，その信号を下位の各 Slave に転送する．また，生存信号から片方の Slave が停止したのを検知した際には，いち早く，もう一方の Slave に停止信号を送信して，脚部の無駄な暴走を防ぐ役割を担っている．また，Master 層の働きにより，12 本ある最下層の信号を６本に絞ることができ，Grand Master にかかる計算負担は半減する．Master の動作は至って単純で，まず，動作１で初期設定．次に動作２で信号送受信開始．最後に動作３で外部信号の変化による，送信動作の適宜変更という動作のみである．

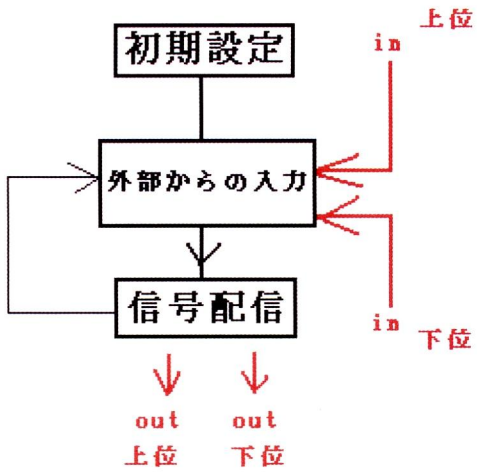


図 32：Master のフローチャート

３．３．４　モータードライバーＡ（SlaveA）

ロボットの脚部の付け根部分に来るモーターを制御するドライバーである．この部分は送信部でモーターを制御するパルス信号と，上位および，SlaveB への生存信号を送信する．また，受信部では，脚部配置決定器(以下，Sub Master)の信号を受けると，パルス波の周期が半周期すれるように設定してあり．この機能によって，どの脚でも自分が右脚であるのか，左脚であるのかを検知し，柔軟に配置に合わせた歩行をする事ができるようする．さらに，SlaveB と生存信号を共有しており，もしも Master が存在しなくても，SlaveB からの生存信号が消えた際には，脚部の動作を停止するように信号を送信する．

SlaveA の動作は，図 33 に示す．まず動作１として，初期設定，次いでタイマー始動が行われ，動作２で各信号送受信が開始される．そして，動作３では一度モーターへの停止信号を受信し，モーターをスタート位置に持ってくるという動作をする．この動作により，機体脚部を初期設定位置に持ってくることで，歩行準備を整える．最後に，動作４でタイマー０割り込み，再び，モーターへのパルス信号を送信開始して，再び動作２へとループする．このような流れで，図 34 に示すようなパルス信号をモーターへ送信する．

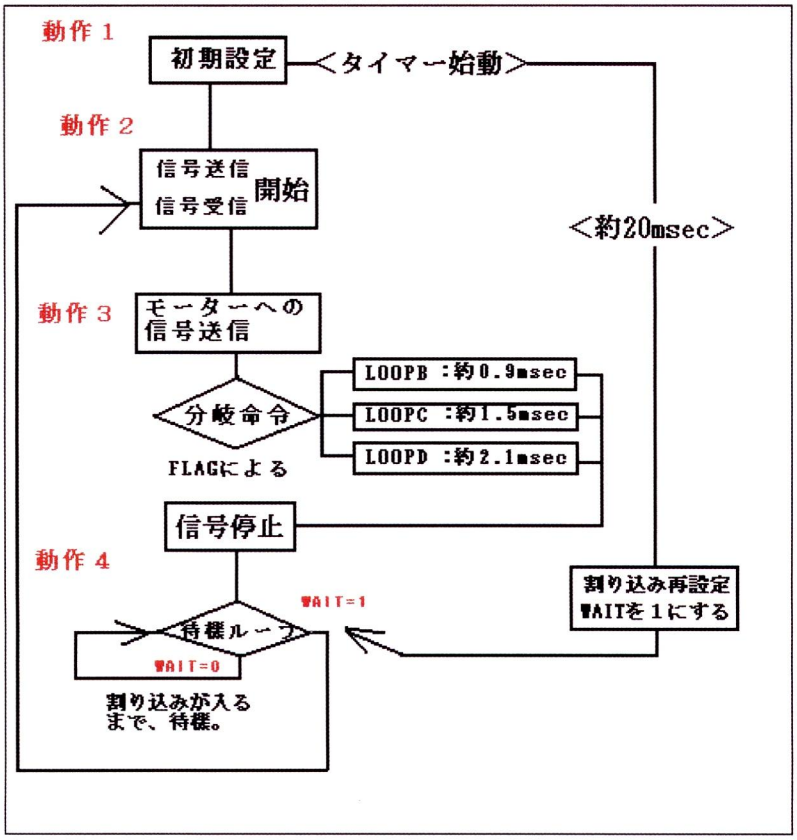


図 33：SlaveA のフローチャート

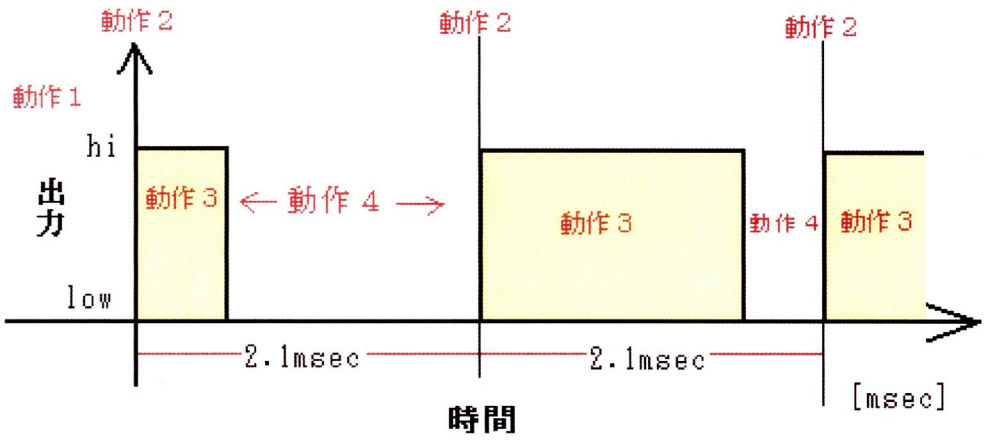


図 34：SlaveA の作り出すパルス波

また，絶えず動作２の部分で，Sub Master からの信号をカウントし，Sub Master からポートＡに信号を受信すると，モーターに送信しているパルス波の位相が半周期，

反転し，モーターの動作も反転するようにプログラムを組む．この機能により，左右の脚を，まったく同じ設計で制作して機体と接続しても，左右の脚が認識でき，さらには，Sub Master の信号制御によって，機体が直進や旋回をする．

３．３．５　モータードライバーＢ(slaveB)

ロボットの脚部膝関節のモーターを制御するドライバーである．この部分は送信部でモーターを制御するパルス信号と，上位および，SlaveA への生存信号を送信している．また，受信部では，脚部配置決定器(以下，Sub Master)の信号を受けると，パルス波の周期が半周期すれるように設定してあり．この機能によって，どの脚でも自分が右脚であるのか，左脚であるのかを検知し，柔軟に配置に合わせた歩行をする事ができるようになっている．さらに，SlaveA と生存信号を共有しており，もしも Master が存在しなくても，SlaveA からの生存信号が消えた際には，脚部の動作を停止するように信号を送信する．

SlaveB の動作は，図 35 に示す．まず動作１として，初期設定，次いでタイマー始動が行われ，動作２で各信号送受信が開始される．そして，動作３では一度モーターへの停止信号を受信し，モーターをスタート位置に持ってくるという動作をする．この動作により，機体脚部を初期設定位置に持ってくることで，歩行準備を整える．最後に，動作４でタイマー０割り込み，再び，モーターへのパルス信号を送信開始して，再び動作２へとループする．そのような流れで，図 36 に示すようなパルス信号をモーターへ送信する．ただ，SlaveA と異なり，Sub Master からの信号は受信していない．また，SlaveB のパルス波は SlaveA と 1/4 周期，ずれた位相で送信されている．このことにより，脚部は歩行動作を生成している．

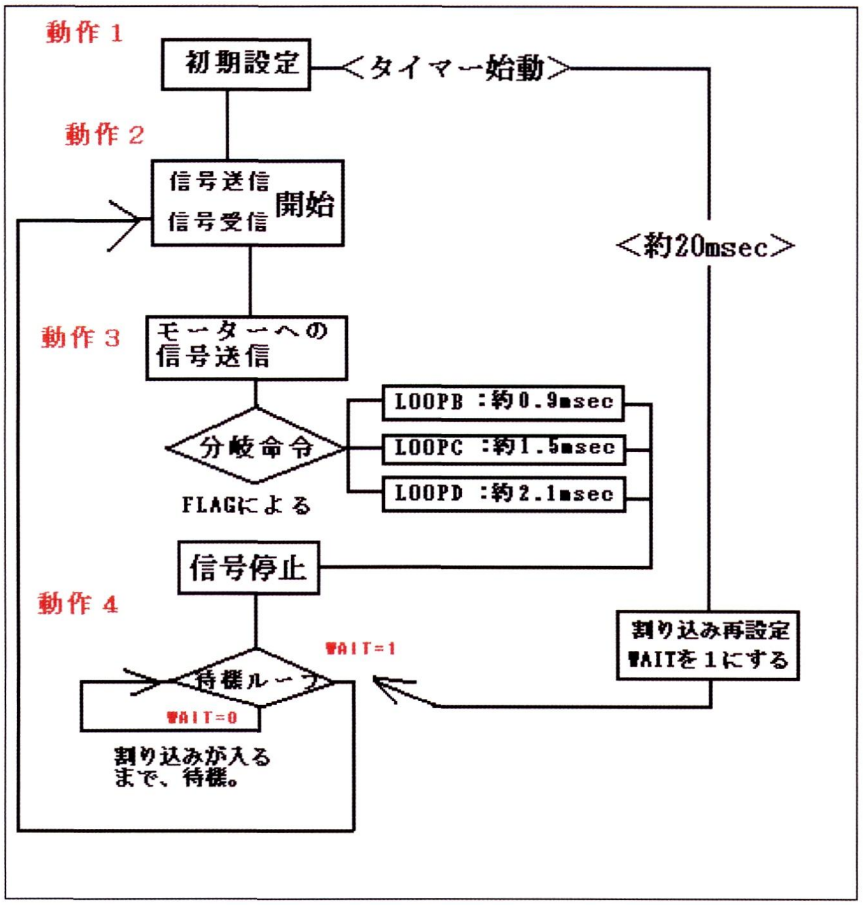


図 35：SlaveB のフローチャート

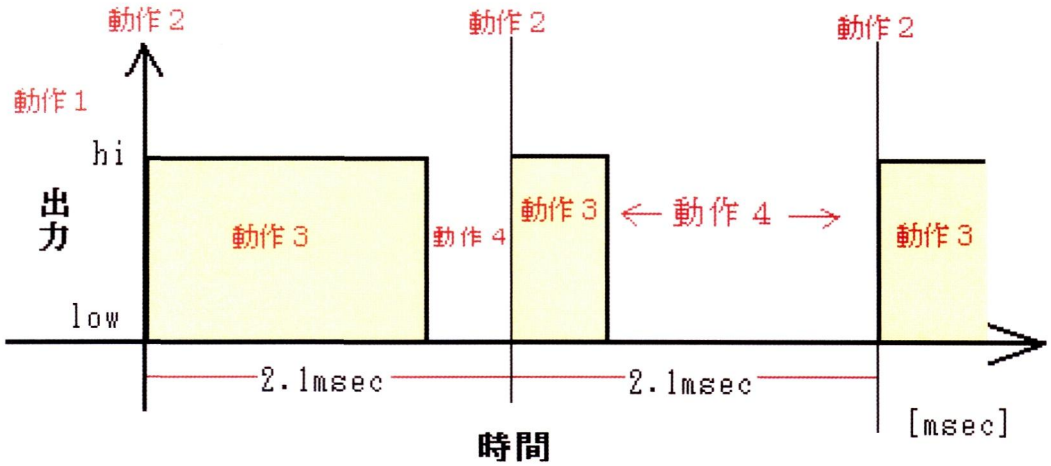


図 36：SlaveB の作り出すパルス波

３．３．６　脚部配置決定器(Sub Master)

パルス波の変調信号を SlaveＡに送信，それを受けた SlaveＡが動作を半周期ずらすことによって，脚部の動作が反転し，ロボット直進をする際の左右の足並みをそろえる事ができる．また，信号の送信対象（任意の SlaveＡ）を変更することによって，ロボットの足並みを変え，転進，バックなどをさせる事により，ロボットの進行方向などを制御する事が可能である．

将来，実機に外部からのミッションを与え，それに応じて動作する際などには，この Sub Master が外部と実機の橋渡しをする事となる．

Sub Master の動作は，図 37 に示す．動作１では，初期設定．動作２では，歩調可変信号を各足の SlaveA に送信となる．動作３は，ミッションによって，歩調可変信号を変化させ，機体を旋回させたり，バックさせたりする際に，SlaveA に送信する信号の ON，OFF を制御するループである．

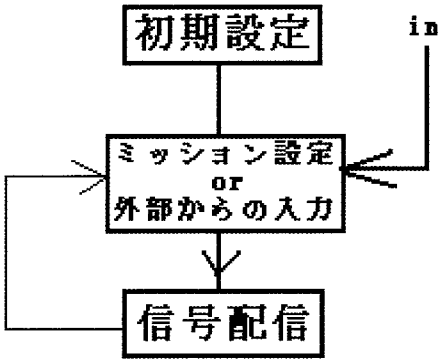


図 37：Sub Master のフローチャート

３．３．７　歩行アルゴリズム

Grand Master のプログラム内には，Walk と言うサブルーチンが存在する．Walk は，Walk0，Walk4，Walk5，Walk6 の４種類に分類でき，それぞれがロボットの歩行パターンを制御している．この Walk 部分では，まず各脚部に一斉送信した停止信号を歩行のタイミングに合わせて解除していき，脚部に歩行パターン通りの動きを行わせている．これら，各サブルーチン Walk についての詳細を以下に示す．

Walk0 は，脚部からの生存信号が３本以下の時に作動する歩行サブルーチンである．脚部が３本以下では，歩行は不能であるので，脚全体に停止信号を送信し，無駄な暴走を防止すると同時に，緊急停止を知らせる信号を送信するルーチンが組む．動作は図 38 に示す．図 38 にある動作５，動作６は，図 31 の動作５，動作６とリンクする．

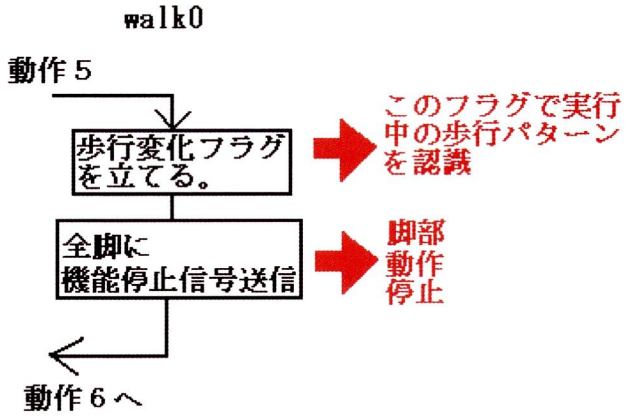


図 38：Walk0 の生成

Walk4 は脚部からの生存信号が４本の時に作動する歩行サブルーチンである．図 39 のように，Ｌ１→Ｒ２→Ｒ３→Ｌ２→Ｌ３→Ｒ１の順で，1/4 動作周期ずつタイミングをずらしながら，残った４本の脚に送っている停止信号を解除して行き歩行を開始させ，機体に波歩行を行わせる．波歩行の動作は，図 39，図 40 に示す．

信号解除は，故障している脚への無駄な処理を防ぐために，故障している脚への信号解除処理をスキップする事で，どの脚が故障しても決まった解除順序とタイミングで停止信号を解除している．そのため，他の歩行パターンよりも，処理が複雑になっている．

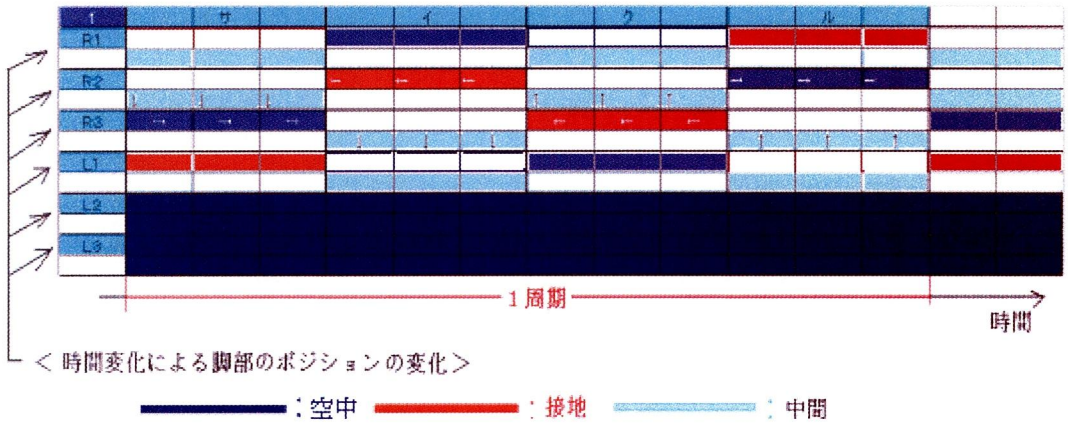


図 39：Walk4(波歩行)の歩行表

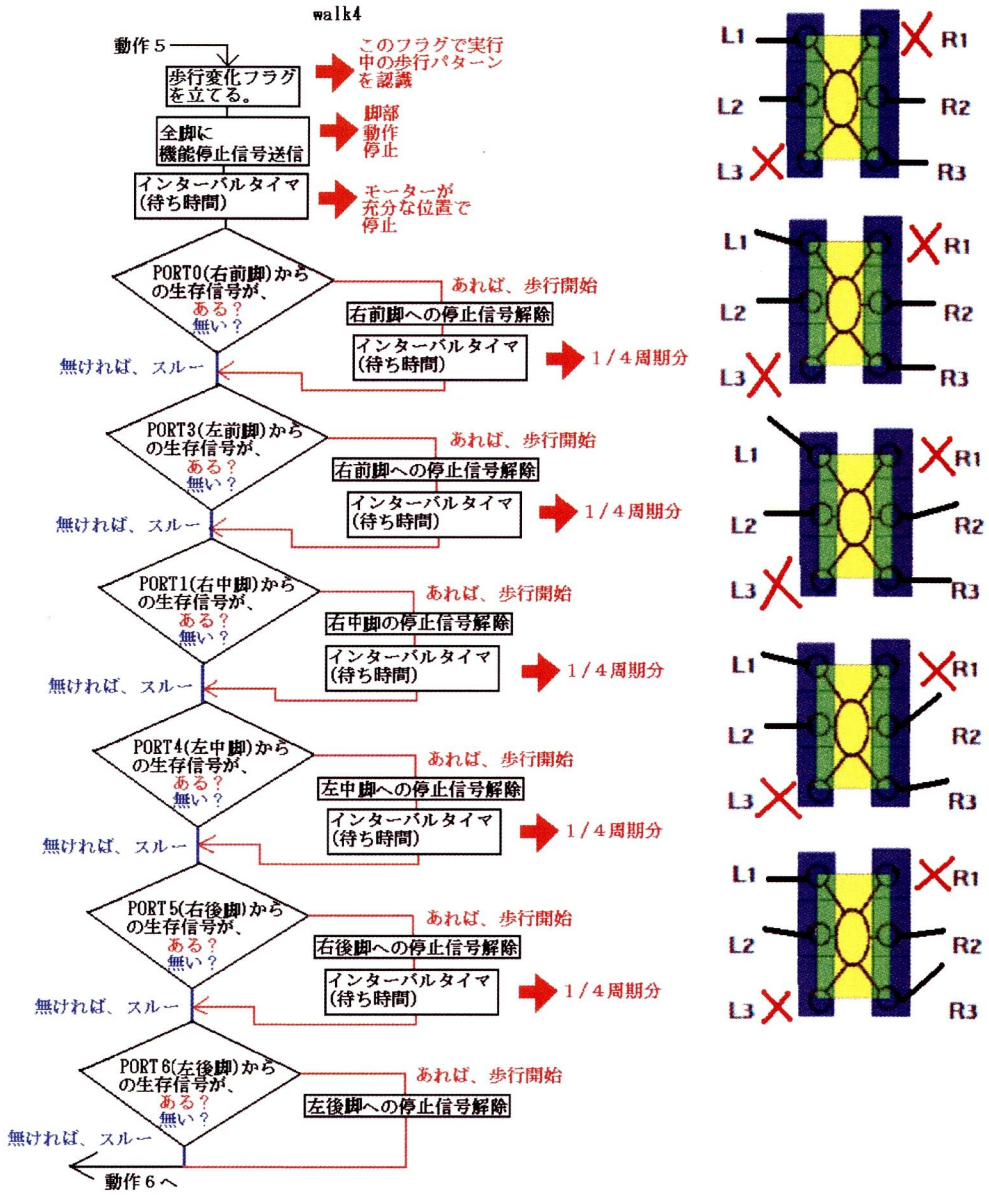


図 **40**：**Walk4** の生成法

Walk5 は，脚部からの生存信号が５本の時に作動する歩行サブルーチンである．残った脚のポジションに関わらず，残った５本の脚を，２脚ずつの３組（１組は１本）に分け，モーターの動作に対して 2/3 周期ずらして停止信号を解除していき，２脚歩行を開始させる． ２脚歩行の歩行表は図 41 に，Walk5 の動作は図 42 に示す．

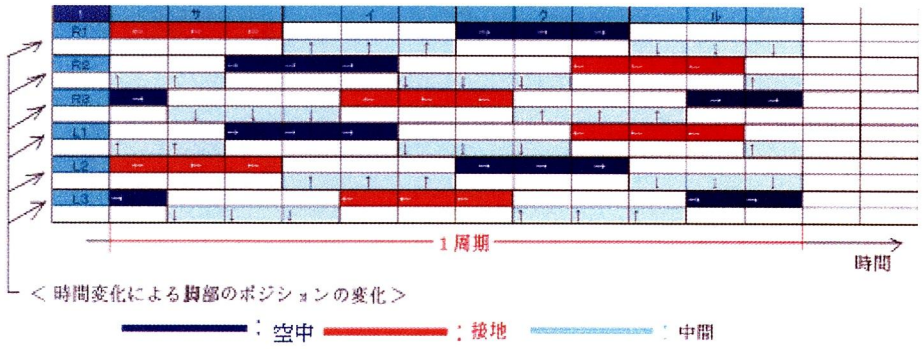


図 41：Walk5(２脚歩行)の歩行表

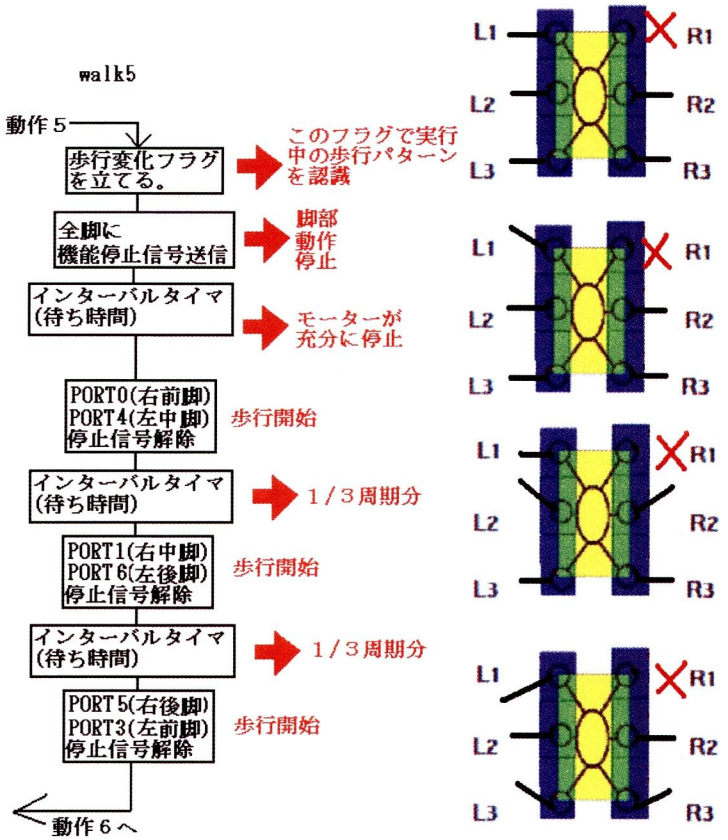


図 42：Walk5(２脚歩行)の生成法

Walk6 は脚部からの生存信号が６本の時に作動する歩行サブルーチンである．６本の脚を３本ずつの２組に分けて，モーターの動作周期に対して 1/2 周期ずらして停止信号を解除していき，歩行を開始させ，機体には３脚歩行を行わせる．３脚歩行は図 43，Walk6 の動作は，図 44 に示す．

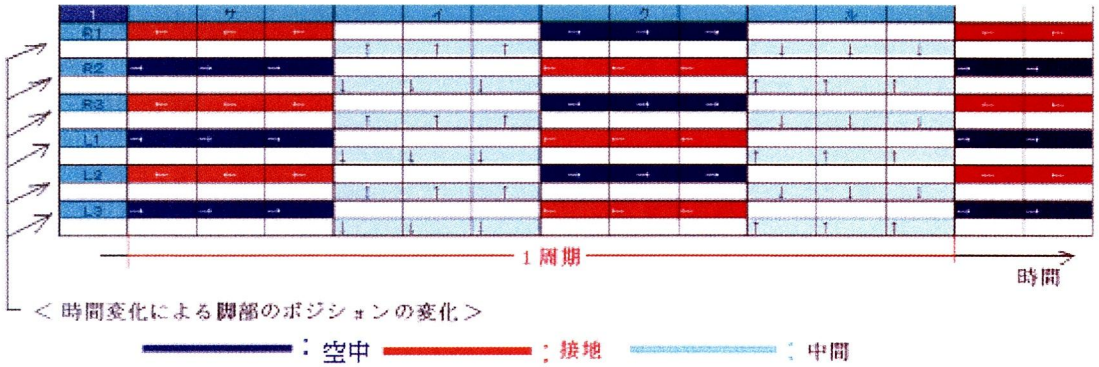


図 43：Walk6(３脚歩行)の歩行表

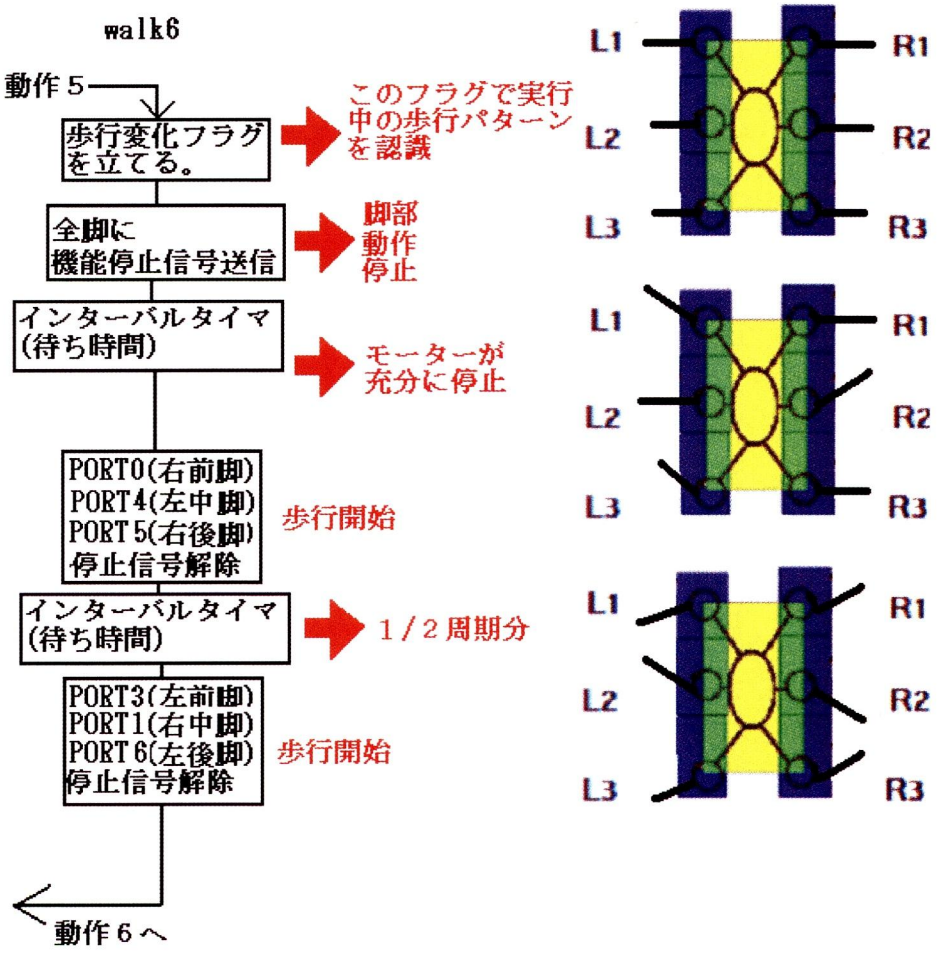


図 44：Walk6(３脚歩行)の生成法

４．実機実験

４．１　目的

完成した機体の冗長性を活かし，故障および破損が機体の運動にどのような影響を及ぼすかを検討する．

４．２　実験方法

本研究で制作した冗長型多足歩行ロボットの脚部を人為的に故障または破損させ，足場を歩行させる．その際の，歩行する時間を計測し，故障または破損が機体に与える影響と，機体の歩行の冗長性と軽い不整地での踏破性を確認する．

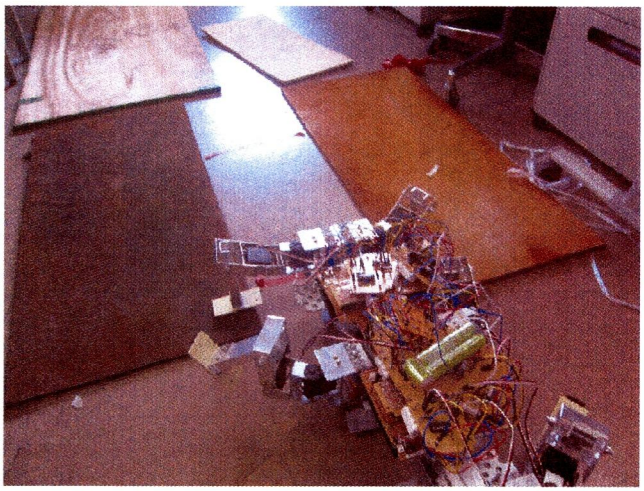


図 45：走行実験の足場

４．３　歩行環境

今回は，高さの若干低い板を数枚ならべ，不整地を再現し，そこで歩行実験を行う．また，スタート地点から，ゴール地点までは 50cm とし，スタートからゴールまでの歩行時間を計測して，歩行の直進性能を比較する．

また，旋回性能に関しては，機体が 180 度旋回するまでの時間を計測し，その時間

を比較することで検討する．旋回し関しては，不整地だと，引っかかりが起き，旋回性能に関わらず旋回時間が延びる事が考えられるため，旋回性の計測は整地で行う．実験に用いた足場は，図 46 に示す．

図 **46**：足場の距離

４．４　歩行データ

表 **2**：計測結果

脚のポジション	前進(秒)	備考	右旋回(秒)	備考
6本	10.44		12.3	
L1故障	14.98	左向きに進む	30.74	
L2故障	23.91	若干、左向き	39.74	旋回半径、大
L3故障	13.41		29.76	
R1故障	17.9	右向きに進む	28.96	
R2故障	28.19	若干、右向き	35.75	旋回半径、大
R3故障	13.07		31.26	
L1破損	13.99	空転多し	15.21	空転多し
L2破損	16.04		20.03	回転半径大
L3破損	8.35		13.57	
R1破損	8.2		17.08	
R2破損	12.02		19.04	回転半径大
R3破損	6.11		15.08	

表 2 の縦の系列は脚のポジションとその状態を示す項目である．６本とは，６本とも健康な脚の時の計測データ．Ｌ１～Ｒ３故障とは，Ｌ（左）の前足（１）～Ｒ（右）の後ろ足（３）の各脚のモーターを停止させた状態での状態を計測したデータ．Ｌ１～Ｒ３破損とは，Ｌ１～Ｒ３の各脚を取り外した状態での計測データをそれぞれ示している．表 7 のデータを前進と旋回に分けてグラフ化すると，以下のようになる．

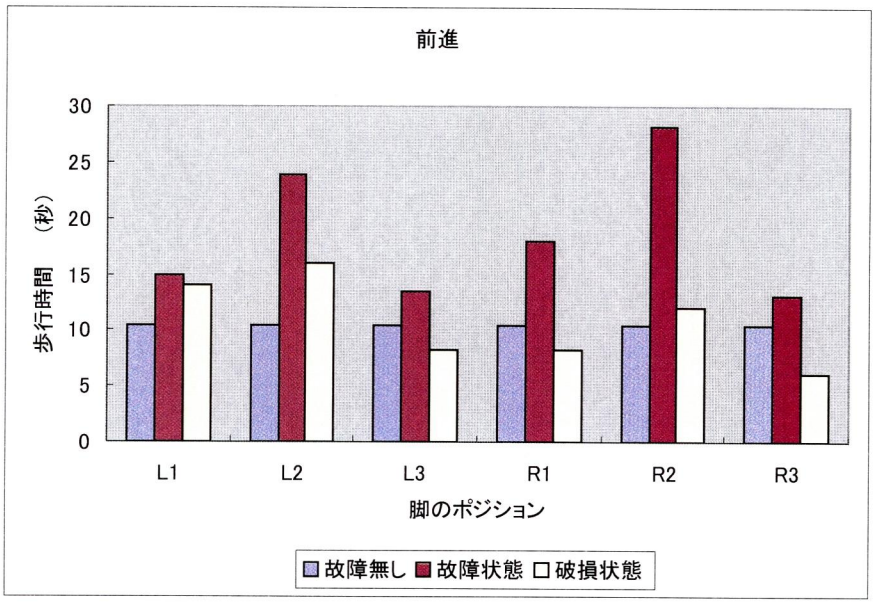


図 47：前進

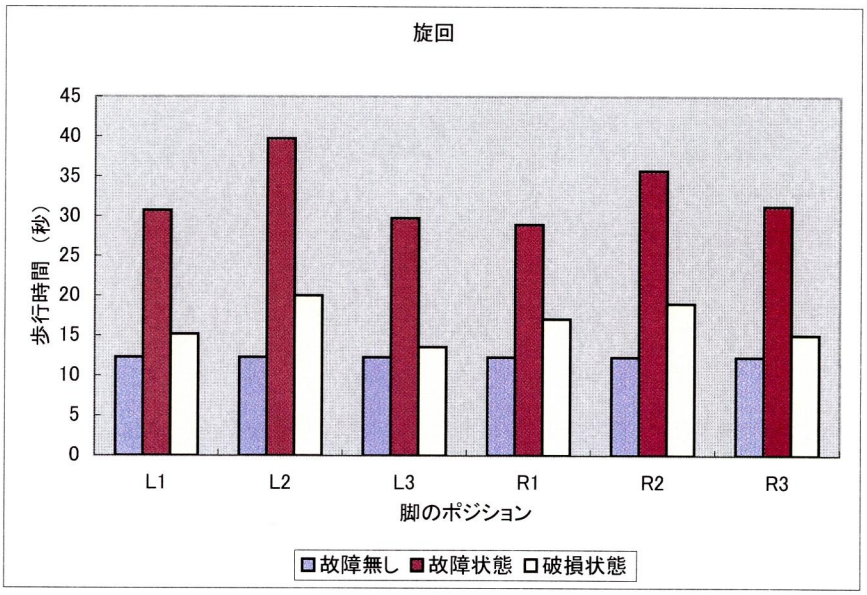


図 48：旋回

4．5　結果

歩行実験では，脚部の破損に対応しデータをとる事ができた．計測結果から，破損と

故障について比較すると，同じ脚部が破損または，故障する場合，直進性，旋回性の両方において，脚部が破損してしまうほうが,良いタイムが出ている．これは，故障状態の脚が，接地してしまい，機体の進行に対してブレーキをかけてしまうためである．また，今回の計測における脚部の破損は，脚部を初期状態の段階から脚部を取り外しているため，重量が軽減されるので，直進時間が短くなったと言える．しかし，実際の現場で脚部が破損した場合は，その脚部が綺麗に外れるとは限らない．脱落した脚部が，他の脚部の動作起動上に落下したり，完全に外れきらず，コードや期待にひっかかって，引きずられる可能性もある．

計測データより，中脚（L2, R2）の重要性が指摘できる．中脚は，前足の次に，歩行の軌道に影響を与える脚なので，中脚が動作不良に陥ると旋回性が大きく落ち込む傾向が見られた．また，直進性においても，中脚を失うと，機体の後ろ脚の空転が目立ち，直進性が落ちた．これは，残っている中脚が不整地に乗り上げた時に，後ろ足の片側が空転し，実質４本脚走行になってしまう事が原因である．また，歩行中の脚の動作を見ていると，脚の動作間隔が狭い．よって，中脚は，動作中，常に前後２つの脚と接近状態にあり，もし，歩行中に中脚が脱落した場合は，確実に歩行の障害になる．その点で，今後，不整地での運用を主眼においている多足歩行機は，中脚のロバスト性に力を入れる事が重要な意味を持つ．

後ろ足は，故障，破損において，他のポジションの脚に比べると，影響はすくない．特に，破損などは，実際の運用でも，他の脚に与える影響が少ない．

5．結言

今回の研究では，どの足が故障し，また，破損しても対応しうる，可変歩行アルゴリズムを生成し実機に搭載することによって，6足歩行ロボットの冗長化に成功した．また，その性能を活かして，脚部の故障と破損が6脚歩行ロボットにどのような影響を与えるかを，検討した．その結果，6脚歩行ロボットの脚部故障が，機体に与える影響は，脚のポジションによって大きく違う事が解った．よって，その違いを踏まえ，脚部の各ポジションのロバスト性を調整する事が，冗長型多足歩行ロボットのさらなる冗長化につながると考えられる．

動作を伴う実験では，機体の放熱が不十分であり，時々，回路がオーバーヒートする事があったが，機体の構造上，回路および機体の分解が容易であったため故障箇所を探し修理する事が比較的容易であった．また，この機体は同一のパーツを多く使用して，組み上げているため，故障時の故障部分を探す際には，故障しているであろう，電子パーツと，正常に動作する回路パーツを比較して故障部分を探す事ができるため，故障部分を探すに当たっても，比較的容易に行え，メンテナンス性の高さも実感することができた．

また，本研究で扱った機体は，Sub Master 部分の設定によっては，外部信号から機体の運動方向などを操作する事ができる．よって，その機能を使用する事で，将来的には実際にミッションをこなす事も可能であると考える．

参考文献

[1] 湯浅秀男　福田敏男　；自律分散制御　；共立出版

[2] TecRobot 工作室；http://homepage1.nifty.com/rikiya/

[3] ニューラルネットワーク入門
；http://mars.elcom.nitech.ac.jp/java-cai/neuro/menu.html

[4] 自律分散型多脚歩行ロボットシステム
；http://www.arai.pe.u-tokyo.ac.jp/research/intro02/leg-j.htm

[5] 神経振動子ネットワーク
；http://staff.aist.go.jp/k.miyashita/projects/ok-semi/sld008.htm

[6] 浅草ギ研　；　自作ロボット入門　；浅草ギ研

[7] 船倉一郎，土屋堯，堀桂太郎；ロボット制御のエレクトロニクス；Ohmsha

[8] 後閑哲也；電子工作のための PIC16F 活用ガイドブック；技術評論社

[9] 田村製作所：http://www.murata.co.jp/ceralock/index.html

謝辞

本研究を行うにあたり，ご指導ご鞭撻を頂いた，伊藤雅則教授，清水悦郎助教授に深く感謝の意を表します．また，本研究に多大なる御協力をして下さった，ロボット工学研究室の方々，機械工作の際にご協力くださった，技官の田端先生にも深く感謝の意を表します．

付録

各ＰＩＣに内臓したプログラムを付録とする.

＜SlaveA＞

```
;slaveA3
;*****************************************
                LIST    P=PIC16F819             ;INIT_set
                INCLUDE         "P16F819.INC"
;__CONFIG_HS_OSC&_WDT_OFF&PWRTE_ON
;*****************************************
        COUNT           EQU             21H             ;count value
        WAIT            EQU             22H             ;
        FLAG            EQU             23H             ;
        LATE            EQU             24H             ;Late Time
        TIME            EQU             25H             ;
;*****************************************
                        ORG             0
                        GOTO    MAIN
                        ORG             4
                        GOTO    INTR
;*************************************************************
MAIN
        ;start set
                        BSF             STATUS,RP0              ;bank1
                        MOVLW           087H
                        MOVWF           OPTION_REG              ;priscaler
        ;timer0 set
                        BCF             STATUS,RP0              ;bank0
                        MOVLW           03CH                    ;time count value
                        MOVWF           TMR0                    ;timer start !
                        BSF             INTCON,TMR0IE           ;timer0 interrupt
                        BSF             INTCON,GIE              ;all        interrupt
able !
        ;ADCON set
                        BSF             STATUS,RP0              ;bank1
```

```
                    MOVLW       06H                     ;ADcon
value
                    MOVWF       ADCON1                  ;all portA
is D-mode
        ;PORTA set
                    MOVLW       0FFH
                    MOVWF       TRISA           ;all portA is input
        ;PORTB set
                    CLRF   TRISB            ;all portB is output
        ;FLAG set
                    BCF         STATUS,RP0      ;bank0

                    MOVLW       000H
                    MOVWF       FLAG            ;FLAG crear
        ;COUNT set
                    MOVLW       0FAH
                    MOVWF       COUNT           ;count  value  is
250
        ;TIME set
                    MOVLW       04H                     ;LOOP
TIMES by 4
                    MOVWF       TIME
        ;LateTime set
                    MOVLW       000H
                    MOVWF       LATE            ;LATE
        ;LifeSignal ON
                    MOVLW       0C0H            ;PORTB 6,7
                    MOVWF       PORTB           ;Life Output ON
;*********************************************************
;*****************************************************
        BML;Before Move Loop
*
                    BTFSS  PORTA,0                  ;Life  Input  Same
*
                    GOTO   $-1                      ;
                *
```

```
                                                                    ,
                    *
                    BTFSC  PORTA,1                  ;Stop Input Over *
                    GOTO   $-1                      ;
                *
;*****************************************************
;*************************************************************
CENTER                                                          ;pulse
input
                    DECFSZ      TIME
                    GOTO   AML

                    MOVLW       04H
                    MOVWF       TIME                ;stop loop times
                    BTFSC  LATE,5               ;late flag
                    CALL   CHANGEFLAG
;*****************************************************
        AML;After Move Loop
         *
                    BTFSS  PORTA,0                  ;Life  Input  Same
*
                    GOTO   MOTION0                  ;
                *
                                                                ;
                    *
                    BTFSC  PORTA,1                  ;Stop Input Over *
                    GOTO   MOTION0                  ;
                *
;*****************************************************
                    BTFSS  PORTA,2
                    GOTO   MOTION1                  ;Right Foot
                    GOTO   MOTION2                  ;Left   Foot
        BACK
                    BCF         WAIT,0              ;wait flag=0

CENTERLOOP
```

```
                BTFSS  WAIT,0                ;if flag=1 -> next skip
                GOTO   CENTERLOOP
                GOTO   CENTER

;*************************************************************
MOTION0                                                      ;STOP
MOTION

                BSF            PORTB,0                       ;RB0 plus
ON

                CLRF   LATE
                CLRF   FLAG
                MOVLW          04H
                MOVWF          TIME
                GOTO   LOOPC                 ;1.5msec
;************************************
MOTION1                                                      ;Right
Foot
                BSF            PORTB,0                       ;RB0 plus
ON

                BTFSS  LATE,5
                GOTO   LOOPC                 ;1.5msec
                BTFSS  FLAG,4
                GOTO   LOOPD                 ;2.1msec
                BTFSS  FLAG,5
                GOTO   LOOPB                 ;0.9msec
                GOTO   LOOPB                 ;0.9msec

;************************************
MOTION2                                                      ;Left
Foot
                BSF            PORTB,0                       ;RB0 plus
ON
```

```
                    BTFSS  LATE,5
                    GOTO   LOOPC              ;1.5msec
                    BTFSS  FLAG,4
                    GOTO   LOOPB              ;0.9msec
                    BTFSS  FLAG,5
                    GOTO   LOOPD              ;2.1msec
                    GOTO   LOOPD              ;2.1msec

;************************************
;************************************************
PULSOFF
                    MOVLW       0FAH
                    MOVWF       COUNT                 ;Counter reset
                    BCF         PORTB,0                     ;plus
OFF

                    GOTO   BACK

;************************************************
;****************************************
        LOOPB
                    NOP                                     ;1*250
                    NOP                                     ;1*250
                    NOP                                     ;1*250
                    NOP                                     ;1*250
                    NOP                                     ;1*250

nop
nop
nop                 NOP                                     ;1*250
                    DECFSZ      COUNT,F                     ;1*249+2
                    GOTO   LOOPB              ;2*249
                    GOTO   PULSOFF                  ;2-> total 2250+2

        LOOPC
                    NOP                                     ;1*250
```

```
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                DECFSZ          COUNT,F                 ;1*249+2
                GOTO    LOOPC                   ;2*249
                INCF    LATE                    ;LATE+1
                GOTO    PULSOFF                         ;2-> total 3750+1

LOOPD
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                DECFSZ          COUNT,F                 ;1*249+2
```

```
                GOTO   LOOPD                ;2*249
                GOTO   PULSOFF                      ;2-> 5250+1
;*****************************************
CHANGEFLAG
                INCF   FLAG,F               ;Flag change
                RETURN
;*****************************************
INTR
                BCF            INTCON,TMR0IF        ;timer0  interrupt
enable
                MOVLW          03CH                 ;timer0 value
                MOVWF          TMR0                 ;timer0 ReStart

                BSF            WAIT,0               ;wait Flag ON
                RETFIE                              ;return
;**********************************

END
```

<SlaveB>

```
;slaveB3
;*****************************************
                LIST   P=PIC16F819          ;INIT_set
                INCLUDE        "P16F819.INC"
;__CONFIG_HS_OSC&_WDT_OFF&PWRTE_ON
;*****************************************
        COUNT          EQU             21H             ;count value
        WAIT           EQU             22H             ;
        FLAG           EQU             23H             ;
        LATE           EQU             24H             ;Late Time
        TIME           EQU             25H             ;
;*****************************************
                ORG            0
                GOTO   MAIN
                ORG            4
```

```
                    GOTO   INTR
;*****************************************
MAIN
        ;start set
                    BSF         STATUS,RP0          ;bank1
                    MOVLW       087H
                    MOVWF       OPTION_REG          ;priscaler
        ;timer0 set
                    BCF         STATUS,RP0          ;bank0
                    MOVLW       03CH                ;time count value
                    MOVWF       TMR0                ;timer start !
                    BSF         INTCON,TMR0IE       ;timer0 interrupt
                    BSF         INTCON,GIE          ;all        interrupt
able !
        ;ADCON set
                    BSF         STATUS,RP0          ;bank1
                    MOVLW       06H                         ;ADcon
value
                    MOVWF       ADCON1                      ;all portA
is D-mode
        ;PORTA set
                    MOVLW       0FFH
                    MOVWF       TRISA               ;all portA is input
        ;PORTB set
                    CLRF   TRISB                ;all portB is output
        ;FLAG set
                    BCF         STATUS,RP0          ;bank0

                    MOVLW       000H
                    MOVWF       FLAG                ;FLAG crear
        ;COUNT set
                    MOVLW       0FAH
                    MOVWF       COUNT               ;count   value   is
250
        ;TIME set
                    MOVLW       04H
```

```
                    MOVWF       TIME                ;Loop times by

        ;LateTime set
                    MOVLW       000H
                    MOVWF       LATE                ;Late Loop
        LifeSignal
                    MOVLW       0C0H                ;PORTB 6,7
                    MOVWF       PORTB               ;Life Output ON
;*****************************************************
;*****************************************************
        BML;Before Move Loop
*
                    BTFSS  PORTA,0                  ;Life  Input  Same
*
                    GOTO   $-1                      ;
                *
                                                            ;
                     *
                    BTFSC  PORTA,1                  ;Stop Input Over *
                    GOTO   $-1                      ;
                *
;*****************************************************
;*****************************************************
CENTER                                                      ;pulse
input
                    DECFSZ      TIME
                    GOTO   AML

                    MOVLW       04H
                    MOVWF       TIME                ;Loop times by 4
                    BTFSC  LATE,7
                    CALL   CHANGEFLAG
;*****************************************************
        AML;After Move Loop
         *
                    BTFSS  PORTA,0                  ;Life  Input  Same
```

```
*
                    GOTO   MOTION0                  ;
               *
                                                    ;
                *
                    BTFSC  PORTA,1                  ;Stop Input Over *
                    GOTO   MOTION0                  ;
               *
;*************************************************
nop
nop
                    GOTO   MOTION                   ;right
        BACK
                    BCF          WAIT,0             ;wait flag=0
CENTERLOOP
                    BTFSS  WAIT,0           ;if flag=1 -> next skip
                    GOTO   CENTERLOOP
                    GOTO   CENTER

;*****************************
MOTION0                                             ;Puls
Control

                    BSF          PORTB,0            ;RB0 plus
ON

                    CLRF   LATE
                    CLRF   FLAG
                    MOVLW        04H
                    MOVWF        TIME
                    GOTO   LOOPC
;*****************************
MOTION                                              ;Puls
Control
                    BSF          PORTB,0            ;RB0 plus
ON
```

```
                    BTFSS  LATE,7
                    GOTO   LOOPC              ;1.5msec
                    BTFSS  FLAG,4
                    GOTO   LOOPD              ;2.1msec
                    BTFSS  FLAG,5
                    GOTO   LOOPB              ;0.9msec
                    GOTO   LOOPB              ;0.9msec
;*********************************
PULSOFF
                    MOVLW       0FAH
                    MOVWF       COUNT                 ;Counter reset
                    BCF         PORTB,0                       ;plus
OFF

                    GOTO   BACK

;*********************************

        LOOPB
nop
nop
nop
                    NOP                                       ;1*250
                    NOP                                       ;1*250
                    NOP                                       ;1*250
                    NOP                                       ;1*250
                    NOP                                       ;1*250
                    NOP                                       ;1*250
                    DECFSZ      COUNT,F                       ;1*249+2
                    GOTO   LOOPB              ;2*249
                    GOTO   PULSOFF                  ;2-> total 2250+2

        LOOPC
                    NOP                                       ;1*250
                    NOP                                       ;1*250
```

```
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                DECFSZ      COUNT,F                     ;1*249+2
                GOTO  LOOPC                 ;2*249
                INCF  LATE                  ;LATE+1
                GOTO  PULSOFF                       ;2-> total 3750+1

LOOPD
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                NOP                                     ;1*250
                DECFSZ      COUNT,F                     ;1*249+2
                GOTO  LOOPD                 ;2*249
```

```
                        GOTO   PULSOFF                    ;2-> 5250+1
;*****************************************
CHANGEFLAG
                        INCF   FLAG,F              ;Flag change
                        RETURN
;*****************************************
INTR
                        BCF            INTCON,TMR0IF       ;timer0  interrupt
enable
                        MOVLW          03CH                ;timer0 value
                        MOVWF          TMR0                ;timer0 ReStar
                        BSF            WAIT,0              ;wait Flag ON
                        RETFIE                             ;return
;*********************************

END
```

<Master>

```
;masterxxx
;*********************************
        LIST           P=PIC16F873
        INCLUDE                "P16F873.INC"
;__CONFIG_HS_OCS&_WDT_OFF&_PWRTE_ON
;*********************************
                        ORG            0
;                       GOTO   MAIN

;************************************************************
MAIN
                                BSF            STATUS,RP0          ;bank 1

                                MOVLW          080H
                                MOVWF          OPTION_REG          ;PORTB
is not pull up

                                MOVLW          06H
```

```
        ;A/D setting value
                                MOVWF       ADCON1
        ;all PORTAs are digital mode

                                MOVLW       0FFH
                                MOVWF       TRISA                   ;all portA
is input

                                CLRF   TRISB                    ;all    portB    is
output

                                BCF         STATUS,RP0              ;bank 0

                                CLRF   PORTA                    ;portA crear
                                CLRF   PORTB                    ;portB crear
;*************************************************************
START
                StartSignal
                                MOVLW       00DH
        ;b'00001101'
                                MOVWF       PORTB
        ;LifeSignal   StopSignal A   StopSignal B

;************
                BML;BeforMoveLOOP
                                BTFSC  PORTA,0
        ;ComparatorSignal
                                GOTO   STOP1

                                BTFSC  PORTA,1
        ;StopSignal
                                GOTO   STOP1

                                BTFSS  PORTA,2
        ;LifeSignal A
                                GOTO   STOP1
```

```
                    BTFSS  PORTA,3
        ;LifeSignal B
                    GOTO   STOP1

                    BSF            PORTB,0
                    BCF            PORTB,1
                    BCF            PORTB,2
                    BCF            PORTB,3

                    GOTO   BML

;***************
STOP1
                    BCF            PORTB,0
                    BSF            PORTB,1
                    BSF            PORTB,2
                    BSF            PORTB,3

                    BTFSC  PORTA,0
                    GOTO   STOP1

                    BTFSC  PORTA,1
                    GOTO   STOP1

                    BTFSS  PORTA,2
                    GOTO   STOP1

                    BTFSS  PORTA,3
                    GOTO   STOP1

                    GOTO   BML
;**************
END
```

＜Grand Master＞

```
;GMx2
;*********************************
        LIST            P=PIC16F873
        INCLUDE                 "P16F873.INC"
;__CONFIG_HS_OCS&_WDT_OFF&_PWRTE_ON
;*********************************

        LEG0            EQU             22H
        LEG4            EQU             23H
        LEG5            EQU             24H
        LEG6            EQU             25H

        COUNT           EQU             26H

        FLAG            EQU             27H
        TIMES           EQU             28H
;************************************************************
                        ORG             0
                        GOTO    MAIN

                        ORG             4
                        GOTO    INTR
;************************************************************
MAIN

;start set

        ;Bank Change
                        BSF             STATUS,RP0              ;bank1

                        MOVLW           087H
                        MOVWF           OPTION_REG              ;priscaler

        ;timer0 set
                        BCF             STATUS,RP0              ;bank0
                ;       MOVLW           03CH                    ;time count value
```

```
                ;       MOVWF   TMR0            ;timer start !
                        BSF     INTCON,T0IE     ;timer0 interrupt
                        BSF     INTCON,GIE      ;all    interrupt
able !

        ;ADCON set
                        BSF     STATUS,RP0      ;bank1
                        MOVLW   06H                     ;ADcon
value
                        MOVWF   ADCON1                  ;all portA
is D-mode



        ;TRISA set
                        MOVLW   0FFH
                        MOVWF   TRISA           ;all portA is input

        ;TRISB set
                        CLRF    TRISB           ;all portB is output

        ;Bank Change
                        BCF     STATUS,RP0      ;bank0


        ;All Flag Reset

                        CLRF    LEG0
                        CLRF    LEG4
                        CLRF    LEG5
                        CLRF    LEG6
                        CLRF    COUNT
                        CLRF    FLAG
                        CLRF    TIMES

;                       CLRF    PORTA           ;portA crear
```

```
                        CLRF    PORTB                   ;portB crear
;*************************************************************
START

        StopSignalStart
                        MOVLW           03FH
        ;b'00111111'
                        MOVWF           PORTB                   ;Life
Signal Stop Signal A   StopSignal B
;************************
        LTL             ;LateTimeLoop 16Loop

                        MOVLW           03CH
;MOVLW          0FFH
                        MOVWF           TMR0                    ;TIMER
start

                        BTFSS   FLAG,0
                        GOTO    $-1
                        CLRF    FLAG                    ;RESET FLAG
                        INCF    TIMES
                        BTFSS   TIMES,4                 ;16
LOOP LATE
                        GOTO    LTL
                        CLRF    TIMES                   ;RESET TIMES
;************************
;test signal
;       MOVLW           03FH
;       MOVWF           PORTA
;************************
        FirstState      ;FIRST STATE CHECK

                        BTFSC   PORTA,0                 ;PORTA's
Signal check
                        CALL    COUNTER                 ;If TEMP
0bit get input
```

```
                BTFSC  PORTA,1                ;1bit
                CALL   COUNTER

                BTFSC  PORTA,2                ;2bit
                CALL   COUNTER

                BTFSC  PORTA,3                ;3bit
                CALL   COUNTER

                BTFSC  PORTA,4                ;4bit
                CALL   COUNTER

                BTFSC  PORTA,5                ;5bit
                CALL   COUNTER

;***********************
        WALK
                INCF   COUNT          ;COUNT + 1

                DECF   COUNT
                BTFSC  STATUS,Z       ;The namber of Legs check
                GOTO   WALK0          ;If Leg is 0

                DECF   COUNT
                BTFSC  STATUS,Z
                GOTO   WALK0          ;If Leg is 1

                DECF   COUNT
                BTFSC  STATUS,Z
                GOTO   WALK0          ;If Leg is 2

                DECF   COUNT
                BTFSC  STATUS,Z
                GOTO   WALK0          ;If Leg is 3
```

```
                    DECF   COUNT
                    BTFSC  STATUS,Z
                    GOTO   WALK4          ;If Leg is 4

                    DECF   COUNT
                    BTFSC  STATUS,Z
                    GOTO   WALK5          ;If Leg is 5

                    DECF   COUNT
                    BTFSC  STATUS,Z
                    GOTO   WALK6          ;If Leg is 6

                    GOTO   OVER           ;OVER LIFE SIGNAL
;*************************************************
;*************************************************
WALK0
                          BSF             LEG0,0                ;FLAG
ON

                          MOVLW           03FH
        ;b'00111111'
                          MOVWF           PORTB                 ;STOP
signal

                SSSC0  ;Stop Signal Controll 0
;****
;none
;****
                          CLRF   COUNT                  ;RESET COUNT
                          GOTO   STATE
;*********************************************
WALK4
                          BSF             LEG4,0                ;FLAG
ON

                          MOVLW           03FH
```

```
        ;b'00111111'
                                MOVWF       PORTB                   ;STOP
signal

                SSSC4   ;Stop Signal Signal Controll 4
;****
                                        BTFSC  PORTA,0
                                        CALL   STEP1

                                        BTFSC  PORTA,3
                                        CALL   STEP4

                                        BTFSC  PORTA,1
                                        CALL   STEP2

                                        BTFSC  PORTA,4
                                        CALL   STEP5

                                        BTFSC  PORTA,2
                                        CALL   STEP3

                                        BTFSC  PORTA,5
                                        CALL   STEP6
;****
                                CLRF   COUNT                    ;RESET COUNT
                                GOTO   STATE
;*************************************************
WALK5
                                BSF         LEG5,0              ;FLAG
ON

                                MOVLW       03FH
        ;b'00111111'
                                MOVWF       PORTB               ;STOP
signal
```

```
                SSSC5   ;Stop Signal Signal Controll 5
;****
                                BCF             PORTB,0
                                BCF             PORTB,4

                                CALL    sub5

                                BCF             PORTB,1
                                BCF             PORTB,5

                                CALL    sub5

                                BCF             PORTB,2
                                BCF             PORTB,3

;****
                                CLRF    COUNT                   ;RESET COUNT
                                GOTO    STATE
;*****************************************************
WALK6
                                BSF             LEG6,0                  ;FLAG
ON

                                MOVLW           03FH
        ;b'00111111'
                                MOVWF           PORTB                   ;STOP
signal
;****
                SSSC6   ;Stop Signal Signal Controll 6

                                MOVLW           015H
                                MOVWF           PORTB
;***************
        sub6
                        MOVLW           03CH
;MOVLW                  0FFH
```

```
                    MOVWF       TMR0                ;TIMER start

                    BTFSS  FLAG,0
                    GOTO   $-1
                    CLRF   FLAG
                    INCF   TIMES
                    BTFSS  TIMES,3                  ;half T   (8LOOP)
                    GOTO   sub6

                    CLRF   TIMES
;***************
                        CLRF   PORTB
                        CLRF   COUNT                ;RESET COUNT
                        GOTO   STATE
;*************************************************
;*************************************************
sub4
;************
        STEP1
                    BCF         PORTB,0
                    GOTO   sub41

;************
        STEP2
                    BCF         PORTB,1
                    GOTO   sub41


;************
        STEP3
                    BCF         PORTB,2
                    GOTO   sub41

;************
        STEP4
                    BCF         PORTB,3
```

```
                    GOTO   sub41

;***********
        STEP5
                    BCF            PORTB,4
                    GOTO   sub41

;***********
        STEP6
                    BCF            PORTB,5
                    GOTO   sub41

;************************
                sub41
                    MOVLW          03CH
;MOVLW              0FFH
                    MOVWF          TMR0                ;TIMER start

                    BTFSS  FLAG,0
                    GOTO   $-1
                    CLRF   FLAG
                    INCF   TIMES
                    BTFSS  TIMES,1                     ;half T   (2LOOP)
                    GOTO   sub41

                    CLRF   TIMES
                    RETURN
;*****************************************************
;*****************************************************
sub5
                    MOVLW          05H                 ;5
                    MOVWF          TIMES               ;5LOOP
        sub51
                    MOVLW          03CH
;MOVLW          0FFH
                    MOVWF          TMR0
```

```
                BTFSS  FLAG,0
                GOTO   $-1
                CLRF   FLAG                ;FLAG Reset
                DECFSZ         TIMES               ;TIMES-1
                GOTO   sub51

                CLRF   TIMES               ;TIMES Reset

                RETURN
;*****************************************************
;*****************************************************
STATE

                        BTFSC  PORTA,0
                        CALL   COUNTER                  ;0bit

                        BTFSC  PORTA,1
                        CALL   COUNTER                  ;1bit

                        BTFSC  PORTA,2
                        CALL   COUNTER                  ;2bit

                        BTFSC  PORTA,3
                        CALL   COUNTER                  ;3bit

                        BTFSC  PORTA,4
                        CALL   COUNTER                  ;4bit

                        BTFSC  PORTA,5
                        CALL   COUNTER                  ;5bit

;************************

        WALKSTATE
                        INCF   COUNT           ;COUNT + 1
```

```
                DECF   COUNT
                BTFSC  STATUS,Z        ;The namber of Legs check
                GOTO   FLAG0          ;If Leg is 0

                DECF   COUNT
                BTFSC  STATUS,Z
                GOTO   FLAG0          ;If Leg is 1

                DECF   COUNT
                BTFSC  STATUS,Z
                GOTO   FLAG0          ;If Leg is 2

                DECF   COUNT
                BTFSC  STATUS,Z
                GOTO   FLAG0          ;If Leg is 3

                DECF   COUNT
                BTFSC  STATUS,Z
                GOTO   FLAG4          ;If Leg is 4

                DECF   COUNT
                BTFSC  STATUS,Z
                GOTO   FLAG5          ;If Leg is 5

                DECF   COUNT
                BTFSC  STATUS,Z
                GOTO   FLAG6          ;If Leg is 6

                GOTO   OVER           ;OVER LIFE SIGNAL
;*************************************************

;*************************************************
        FLAG0
                        CLRF   LEG4           ;OTHER FLAG CLER
                        CLRF   LEG5
```

```
                    CLRF    LEG6

                    BTFSS   LEG0,0          ;Pass Flag0 Check
                    GOTO    WALK0           ;If Flag0 none
                    GOTO    STATE           ;If Flag0 already have had
;***************
            FLAG4
                    CLRF    LEG0            ;OTHER FLAG CLER
                    CLRF    LEG5
                    CLRF    LEG6

                    BTFSS   LEG4,0          ;Pass Flag4 Check
                    GOTO    WALK4           ;If Flag4 none
                    GOTO    STATE           ;If Flag4 already have had
;***************
            FLAG5
                    CLRF    LEG0            ;OTHER FLAG CLER
                    CLRF    LEG4
                    CLRF    LEG6

                    BTFSS   LEG5,0          ;Pass Flag5 Check
                    GOTO    WALK5           ;If Flag6 none
                    GOTO    STATE           ;If Flag6 already have had
;***************
            FLAG6
                    CLRF    LEG0            ;OTHER FLAG CLER
                    CLRF    LEG4
                    CLRF    LEG5

                    BTFSS   LEG6,0          ;Pass Flag6 Check
                    GOTO    WALK6           ;If Flag6 none
                    GOTO    STATE           ;If Flag6 already have had
;***************
            OVER
                    MOVLW       0FFH
                    MOVWF       PORTB
```

```
                        GOTO   WALKSTATE
                        ;GOTO  $
;*************************************************
;*************************************************
COUNTER
                        INCF   COUNT                    ;Legs Counter
                        RETURN
;*************************************************
;*************************************************
INTR
                        BCF            INTCON,T0IF   ;timer0 interrupt enable

                        BSF            FLAG,0                ;FLAG ON
                        RETFIE                               ;return
;**********************************
;*************************************************
END
```

<Sub Master>

```
;swich
;**********************************
        LIST            P=PIC16F819
        INCLUDE                 "P16F819.INC"
;__CONFIG_HS_OCS&_WDT_OFF&_PWRTE_ON
;**********************************

LEDD1           EQU             00H
LEDD2           EQU             01H
CNT1            EQU             0CH
CNT2            EQU             0DH
CNT3            EQU             0EH
LEDD3           EQU             0FH             ;add

                ORG             0
```

```
                BSF             STATUS,RP0
                CLRF    TRISB
                BCF             STATUS,RP0

REPEAT          MOVLW           07H
                MOVWF           PORTB

                GOTO    $-2
;*******************************************************
                CALL    TIMER3


                MOVLW           LEDD2
                MOVWF           PORTB

                CALL    TIMER3

                GOTO    REPEAT

TIMER1          MOVLW           D'62'
                MOVWF           CNT1
LOOP1 NOP
                DECFSZ          CNT1,1
                GOTO    LOOP1
                RETURN

TIMER2          MOVLW           D'100'
                MOVWF           CNT2
LOOP2 NOP
                CALL    TIMER1
                DECFSZ          CNT2,1
                GOTO    LOOP2
                RETURN

TIMER3          MOVLW           D'100'
                MOVWF           CNT3
```

```
LOOP3 NOP
              CALL   TIMER2
              DECFSZ         CNT3,1
              GOTO   LOOP3
              RETURN

              END
```